TI DAS MAGAZIN FÜR TI PC & TI 99/4A REVUE

Book Collection 3

- Special Edition -

special+special+special+

TI 99/4A SPECIAL

SONDERHEFT Nr. 1/85 DM 12,80

RUND 150 SEITEN LISTINGS! LISTINGS!

ÖS 108/SFR 12,80

ANWENDERPROGRAMME:

Z.B. Pixelmaster/Funktionsplot
Säulen-Diagramme/Sound-Tabellen
Kontokorrentrechnung/Zinseszinsen
Texteditor/Dateiverwaltung

SCHULPROGRAMME SPIELPROGRAMME:

Z.B. Mathe/Physik/Französisch-Vokabeln
Schiffe-Versenken/Monopoly/Mini-Golf
Adventures und Knobel-Spiele

UTILITIES:

Z.B. Betriebs-Systeme/Speeche ditor

special+special+special

NEU NEU NEU NEU NEU NEU NEU NEU

Rausch & Haub

präsentieren:

SOFTWARE + HARDWARE ZUBEHÖR + UTILITIES

für

TI-99/4a

+ SCHNEIDER CPC 464

Hagera® SOFTWARE

Anwendungen, Spiele, Grafik, u. v. m., natürlich auch in TMS 9900 ASSEMBLER

BROKER-BÖRSENSPIEL

mit Sammelordner, ausf. Beschreibung, Diskette für X-Basic/Speichererweit.

nur DM 29.90

ANWENDUNGEN + SPIELE

auf Diskette oder Cassette, für alle Ausbaustufen (auch Grundversion)

schon ab DM 15.90

TOP-SOFTWARE:

TRAFFIC Actionspiel (X-Basic)	DM 24.90
ECONOMY Strategie (XB, 32 Kb)	DM 24.90
SPIELEPAKETE m. 3 0,5 Sp.	je DM 59.90
SUPER GRAFIC SET (XB, 32 Kb)	DM 69.90
BASIC GRAFIC EXP. (Assembl.)	DM 34.90
HARDCOPY f. Epson (Assembl.)	DM 15.90
ASSEMBLER-KURS (in Kürze)	DM 79.—

Daten-/Adressenverwaltungen aller Art sowie viele weitere Spitzenprogramme! (Preise zzgl. Versand)

Vertrieb von:

Original Soft-/Hardware und Zubehör — soweit noch Produktion vorhanden — von diversen Fremdherst./-anbietern. (Keine Lagerhaltung dieser Artikel)

LISTE 80-Pf.-Briefmarke

(Bitte System TI oder CPC angeben!)

GESAMTKATALOG

- 60 Seiten Beschreibungen von ca. 200 Artikeln,
- Tips & Tricks, Angebote für Clubs und Hacker,
- Hintergrund-Story sowie Info über Problemlösungen
- nach Ihren Vorlagen . . . gg. DM 3.50 UKB!
- (Volle Vergütung b. ein. Bestellung)

ANWENDER-SERVICE

- Profi-Anwendungen u. Problemlösungen nach Ihren
- Vorstellungen u. Vorlagen zu günstigen Konditionen für
- Büros, Kaufleute und alle, die selbst keine Zeit zum
- Programmieren haben. Senden Sie uns einfach Ihre
- Unterlagen oder fordern Sie unseren Katalog (s. o.) mit
- weiteren Informationen an.

Wir freuen uns auf Ihre Post:

RAUSCH & HAUB
Vertriebsgesellschaft dbR
POSTFACH 32 03 13 ✻ 5300 BONN 3

TI-99/4 A

PERIPHERIE

TI-99/4 A (Konsole)	349,—
Peripheriebox + Diskettenlaufwerk + Controller	a.A.
P-Code-Karte	799,—
RS 232-Karte (Myarc)	399,—
32 K-Karte	449,—
2. Laufwerk extern (Toshiba Slimline DSDD) mit Gehäuse und Netzteil	949,—
Externes Druckerinterface + Kabel + Epson RX 80	1299,—
dto. + RX 80 FT+	1519,—
dto. + FX 80+	1819,—
dto. + Star SG-10	1279,—
Sprachsynthesizer	189,—
Grafiktablett (Supersketch)	239,—
Modulexpander 3-fach	125,—
8-fach	225,—
Akustikkoppler Dataphon S 21 d + externe V-24-Schnittstelle + Verbindungskabel	579,—
Joystickinterface + 2 Joysticks Quickshot II	99,—
Fernbedienung (Original TI, 2 Joysticks)	65,—

BÜCHER

Editor/Assembler Handbuch dt.	98,—
Extended Basic Handbuch dt.	48,—
Mini Memory Spezial dt.	55,—
Assemblerhandbuch für das Mini Memory dt.	78,—
TI-99/4 A intern in dt. (Betriebssystem, GPL, ROM ...)	38,—

MODULSOFTWARE

Extended Basic (Original TI)	259,—
Extended Basic (dt. Nachbau)	239,—
Editor/Assembler (32 K notw.)	175,—
TI-Writer (32 K notw.)	299,—
Multiplan (32 K notw.)	299,—
Mini Memory	289,—
Terminal Emulator II	95,—
Connect four, Yahtzee, Games II, Rechenkünstler	je 29,—
Zero Zap	je 29,—
Attack, Chisholm Trail, Othello, Tombstone City, TI-Invaders	je 39,—
Parsec, Alpiner, Indoor soccer, Car Wars	je 49,—
Fathom, Microsurgeon, Moonmine, Moonsweeper, Statistik	je 59,—
Espial, Defender, Dig Dug, Tunnels of doom	je 69,—
Espial + Defender + Dig Dug	175,—
Buck Rogers, Burgertime, Congo Bongo, Pirate's Isle, Treasure Island	je 75,—
Moon Patrol, Jungle Hunt, Donkey Kong	je 89,—
Pole Position	99,—
Video Chess	79,—

DISKETTEN- UND CASSETTENSOFTWARE

Superbasic, Exbasic II + Painter, Extended Basic Compiler, Graphicmaster, 3-D-World, Forth, Apesoft, TI-Text, TI-Datei, Skat ...	a.A.

Alle Preise incl. MWSt. zuz. Versandkostenpauschale (Warenwert bis DM 1 000,—/darüber): Vorauskasse (DM 8,—/20,—); Nachnahme (DM 11,20/23,20); Ausland (DM 18,—/30,—). Versand nur gegen Vorauskasse oder per NN; Ausland nur Vorauskasse. Gesamtpreisliste gegen Freiumschlag. Alle Angebote gültig, solange Vorrat reicht.

CSV RIEGERT
Schloßhornstr. 5, 7324 Rechberghausen,
Tel. (0 71 61) 5 28 89

UNSER ☎ TELEFONSERVICE ☎

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf dieser Seite

ACHTUNG!!!

Ab sofort steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33220 und für Listings/Programme: 089/286036.

SONDERHEFT
Nr. 1/85

INHALT

Grüß Gott
Gruezi
Guten Tag

Nach elfmonatigem Bestehen der TI-REVUE können wir Ihnen nun den ersten Sonderband TI-SPECIAL präsentieren. Dank der eifrigen Mitarbeit unserer Leser wird sich in diesem Band für jeden etwas finden lassen. Wohlwissend, daß noch mehr als die Hälfte der TI-User nur über die Grundkonfiguration (Konsole und Kassettenrekorder) verfügt, haben wir diesen Bank so zusammengestellt, daß etwa 50 % der Programme ohne Erweiterungen lauffähig sind.
Selbstverständlich haben wir auch die User berücksichtigt, die ihren Rechner weiter ausgebaut haben und ständig auf der Suche nach geeigneten Programmen für ihre Erweiterungen sind.
In einzelnen Abschnitten geordnet finden Sie Anwenderprogramme, Spiele und Utilities. Alle vorgestellten Programme wurden von uns überprüft. sie sind einwandfrei lauffähig und abbruchsicher. Für welche Konfigurationen diese Programme geeignet sind, ist dem jeweiligen Programmkopf zu entnehmen.
Zum Verständnis dieses Heftes: Wir haben die Programme grob klassifiziert in die Sparten Utilities, Schule, Anwenderprogramme und Spiele. Wobei dies wirklich nur als kleiner Rahmen gedacht ist. Denn: Die Dateiverwaltung aus dem Anwenderbereich kann für ein Utility ebenso wichtig sein wie der Text-Editor für die Schulsparte oder die Tabellen-Erstellung für den Ausbau eines Spieles und vice versa. Der Kombinationsmöglichkeiten gibt es viele.
Nun wünschen wir Ihnen ein Errorfreies Abtippen und viel Vergnügen mit Ihrem TI-SPECIAL.

Ihre Redaktion

UTILITIES SCHULE ANWENDER SPIELE IN TI BASIC

Funktionsplot	16
3.3-Determinante	20–21
Quadr. Ergänzungen	22–23
Französische Zahlen	24–25
Dateiverwaltung	26–28
Tabellen-Erstellung	33–34
Jahreszinsen	35–39
Blumengießen	61–64
Black Hole	65–69
Schiffe versenken	82–85
Pushit	88–92
Codo	100–104
Minigolf	105–108
Hammurabi	109–112
Buddy Jo	136–140

UTILITIES SCHULE ANWENDER SPIELE IN EX BASIC

Erweitertes Betriebs-System	4–8
Der Sprachsatz des Speechsynthesizers	9–12
Pixelmaster	13–15
Sound-Editor	19
Sternenhimmel	17–18
Videothek	29–32
Texteditor	43–44
Kontokorrent-Buchhaltung	40–42
Physik	113–125
Spinnen-Spiel	45–49
Heinrich der Schreckensherrscher	50–54
Vulkanien	55–60
Feierabend	80–81
Vierkampf	70–79
Break Out	86–87
Wortsuchspiel	98–99
Broker	126–135
Monopoly	93–97
Back Gammon	141–146

Impressum

TI-SPECIAL ist eine Sonderausgabe der TI-REVUE. Sie erscheint im TI/CBM-Verlag, Postfach 1107, Elisabethweg 1, 8044 Lohhof.
Redaktion: Senator-Presse-Service.
Verantwortlich für den Inhalt: Harmut König, Werner E. Seibt. Verantwortlich für Anzeigen: Bruno Redase.
Alle: Postfach 1107, 8044 Lohhof.
Anfragen bitte nur schriftlich. Vertrieb: Verlagsunion Wiesbaden. Es gilt die Honorarliste des Verlages. Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung. Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für einen einmaligen Abdruck und die Aufnahme in den Kassetten-Service.

Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Jedwede Verwertung ist untersagt, Nachdruck nur mit ausdrücklicher schriftlicher Zustimmung des Verlages.
© by TI-CBM-Verlag, SPS und Autoren.

Erstverkaufstag dieser Ausgabe: 29. Mai 1985
Es gilt Preisliste Nr. 5 vom 1.1.1985.
Media-Unterlagen bitte anfordern bei Bruno Redase, co TI/CBM Verlag, Postfach 1107, 8044 Lohhof.
Für telefonische Anfragen: 089/28 60 35
Printed in Germany by Mayer & Söhne

BETRIEBSSYSTEM

Das Programm enthält drei Routinen.

1. Delete: Erlaubt, aus einem Programm einen Block von Zeile Z1 – Z2 zu entfernen und das verkürzte Programm unter einem neuen Namen auf Diskette abzuspeichern.
2. Save: Erlaubt, aus einem Programm einen Block von Zeile Z1 – Z2 herauszukopieren und unter einem neuen Namen auf Diskette abzuspeichern.
3. Load Sub: Erlaubt das Aneinanderfügen beliebig vieler Programme, wobei die angefügten Programme automatisch umnumeriert werden (einschließlich Sprungadressen!!). Die Schrittweite ist frei wählbar. Damit entfällt die Gefahr, daß Programmzeilen überschrieben werden, was bei „Merge" ja leicht passiert. Außerdem muß man nicht nach jedem „Merge"-Befehl das Programm mit „Res" umnumerieren. Das neue Programm wird unter einem neuen Namen auf Diskette abgespeichert.

Das Programm biete nach „Run" obiges Auswahlmenü an. Nach Start der Routine wird die Eingabe von Programmname und Diskettenlaufwerk erwartet. Laufwerkbezeichnung 1, 2, 3 wird als Laufwerkname, alle anderen Eingaben als Diskettenname akzeptiert.
Es wird geprüft, ob Sie nicht zwei gleiche Dateien auf einer Diskette eröffnen.
Während der Ausführung wird angezeigt, was der Rechner gerade bearbeitet.
Das Programm belegt 6 541 Byte im Speicher, es können damit theoretisch beliebig lange Programme erzeugt werden.
Wichtig: Alle zu bearbeitenden Programme müssen vorher mit "SAVE DSK ....., MERGE" gespeichert worden sein.
Benötigte Gerätekonfiguration:
TI – 99/4a Konsole
TI Extended Basic
1 Disk-Laufwerk (besser 3 Diskettenlaufwerke)

Peter Heimlich

SPRACH-SYNTHESIZER

Vokabular des SPRACH-SYNTHESIZERS

Wer einen Sprach-Synthesizer besitzt und auch Programme mit Sprachausgabe schreibt, wird dieses Hilfsprogramm zu schätzen wissen. Es ist doch etwas umständlich, sich den Sprachschatz des Synthesizers Wort für Wort im Command-Mode vorsprechen zu lassen. Dieses Programm spricht und schreibt wahlweise den gesamten Sprachschatz, das Alphabet, das Zahlensystem oder andere ausgewählte Bereiche aus dem Vokabular des Sprach-Synthesizers.
Sie können nun in Ruhe die Wörter oder Sätze auswählen, die Sie in Ihrem Programm verwenden möchten. Vielleicht ist dieses Programm auch eine kleine Hilfe für alle, die gerade Englisch lernen. Sie können die Aussprache der englischen Wörter üben. Die Benutzerführung dieses Programmes läßt keinerlei Fragen offen und das Programm ist problemlos zu bedienen.

W. Liebender

PIXEL-MASTER

Der Ti99/4A besitzt keine Befehle, die das Ziehen einer Linie oder das Setzen eines Punktes ermöglichen. Man muß erst mit CHAR die einzelnen Punkte definieren, bevor man sie auf den Bildschirm bringen konnte.
Mit Pixel-Master haben Sie nun die Möglichkeit, Kreise, Bögen, Linien, Geraden, Rechtecke und Punkte zu setzen.
Zum Programmablauf:
Nach der Titelmelodie:
Nach der Titelmelodie werden die einzelnen Variablen erklärt. Dann erscheint das Menue. Wenn Sie eine Figur gewählt haben, werden die Werte dafür abgefragt.
Hier eine genauere Erklärung:

1 : Kreis / Ellipse
X = Spalte des Mittelpunktes
Y = Zeile des Mittelpunktes
R = Senkrechter Radius
R1 = Waagrechter Radius

2 : Kreisbogen
X, Y, R, R1 wie bei Kreis / Ellipse
A = Anfangswinkel (0–359 Grad)
E = Endwinkel (E größer A)

3 : Linie
X = Spalte
Y = Zeile
A = Winkel von X, Y aus
L = Länge der Linie

4 : Gerade
X1 = Spalte 1. Punkt
Y1 = Zeile 1. Punkt
X2 = Spalte 2. Punkt
Y2 = Zeile 2. Punkt

5 : Rechteck
X = Spalkte linke obere Ecke
Y = Zeile linke obere Ecke
M = Länge Kante A
N = Länge Kante B
A = Drehung des Rechtecks

6 : Punkt
X1 = Spalte
Y1 = Zeile

Zum Listing:
Ab Zeile 1160 beginnt das Unterprogramm zum Zeichen der Figuren.

Sascha Nungeß

```
100 ! ************************
110 ! *    ERWEITERTES-      *
120 ! *   BETRIEBSSYSTEM     *
130 ! * Copyright 1984  by  *
140 ! *   Peter HEIMLICH     *
150 ! *  beaerbeitet  vom    *
160 ! *      TEAM  des       *
170 ! *  AKTUELL VERLAGES    *
180 ! *       MUENCHEN       *
190 ! * benoetigte Geraete  *
200 ! *  TI 99/4a KONSOLE    *
210 ! * TI  EXTENDED BASIC  *
220 ! *   mind. 1 DSK/LW     *
230 ! *   max.  3 DSK/LW     *
240 ! *                      *
250 ! *    BENOETIGTER       *
260 ! *   SPEICHERPLATZ      *
270 ! *    6627  BYTES       *
280 ! *                      *
290 ! ************************

300 ON WARNING NEXT
310 CALL PH("BETRIEBSSYSTEM"
,"12/84")
320 E$=CHR$(255)&CHR$(255)
330 T1$="LAUFWERKBEZEICHNUNG
 1, 2, 3 WIRD ALS LAUFWERKNA
ME,......ALLES ANDERE ALS DI
SKETTEN- NAME INTERPRETIERT"
340 T2$="PROGRAMM A......:..
.........LAUFWERK.........:"
350 T3$="PROGRAMM B......:..
.........LAUFWERK.........:..
.........SCHRITTWEITE B..:.1
0"
360 T4$="NEUER NAME......:..
.........LAUFWERK.........:"
370 T5$="ANFANGSZEILE Z1 :..
.........ENDZEILE.....Z2 :"
380 DISPLAY AT(1,1)ERASE ALL
:"ERWEITERTES BETRIEBSSYSTEM
": : :"...>.1..DELETE": :"..
.>.2..SAVE": :"...>.3..LOAD
SUB": :"...>.4..ENDE"
390 DISPLAY AT(13,1):"ZU BEA
RBEITENDE PROGRAMME   MUESSE
N MIT":"SAVE 'DSK......',MER
GE":"GESPEICHERT WORDEN SEIN
 !"
400 DISPLAY AT(18,1):"ALLE W
EITEREN ERKLAERUNGEN..FINDEN
 SIE NACH DER WAHL....EINES
BEFEHLS."
410 DISPLAY AT(22,1)BEEP:"WA
EHLEN SIE"
420 CALL WAHL(4,W)
430 ON W GOSUB 480,560,990,4
70
440 IF L3$="" THEN 380
450 DISPLAY AT(8,1)ERASE ALL
:"ZUR WEITEREN ARBEIT MIT DE
R NEU ERZEUGTEN PROGRAMMDATE
I": :TAB(11);P3$: :"GEBEN SI
E <NEW> EIN UND.....LADEN DI
E PROGRAMMDATEI MIT"
460 DISPLAY AT(16,7):"MERGE
";L3$
470 END
480 ! *** DELETE ***
490 DISPLAY AT(1,1)ERASE ALL
:"DELETE": :"LOESCHT":"PROGR
AMMZEILE  Z1-Z2":"AUS PROGRA
MM   A"
500 DISPLAY AT(8,1):T1$
510 DISPLAY AT(15,4)BEEP:">
1..START": :"...> 2..ZURUECK
"
520 CALL WAHL(2,I)
530 ON I GOTO 540,380
540 DISPLAY AT(1,12)ERASE AL
L:"DELETE": : : :T2$: :T5$:
:T4$
550 GOTO 630
560 ! *** SAVE ***
570 DISPLAY AT(1,1)ERASE ALL
:"SAVE": : :"SPEICHERT":"PRO
GRAMMZEILE..Z1-Z2":"AUS PROG
RAMM...A"
580 DISPLAY AT(8,1):T1$
590 DISPLAY AT(15,4)BEEP:">
1..START": :"...> 2..ZURUECK
"
600 CALL WAHL(2,I)
610 ON I GOTO 620,380
620 DISPLAY AT(1,13)ERASE AL
L:"SAVE": : : :T2$: :T5$: :T
4$
630 ACCEPT AT(5,19)SIZE(-10)
BEEP:P1$
640 IF P1$="" THEN 630
650 ACCEPT AT(6,19)SIZE(-10)
BEEP:L1$
660 IF L1$="" THEN 650
670 CALL DATEI(L1$,P1$)
680 ACCEPT AT(8,19)VALIDATE(
DIGIT)SIZE(-5)BEEP:Z1
690 IF Z1=0 THEN 680
700 ACCEPT AT(9,19)VALIDATE(
DIGIT)SIZE(-5)BEEP:Z2
710 IF Z2<Z1 THEN 680
720 ACCEPT AT(11,19)SIZE(-10
)BEEP:P2$
```

```
730 IF P2$="" THEN 720
740 ACCEPT AT(12,19)SIZE(-10
)BEEP:L2$
750 IF L2$="" THEN 740
760 CALL DATEI(L2$,P2$)
770 IF L1$=L2$ THEN CALL DAT
FEHLER :: GOTO 630
780 P3$=P2$ :: L3$=L2$
790 OPEN #1:L1$,DISPLAY ,VAR
IABLE 163,INPUT
800 OPEN #2:L2$,DISPLAY ,VAR
IABLE 163,OUTPUT
810 DISPLAY AT(16,1):"LESEN"
:"DATEI :":"ZEILE : >"
820 DISPLAY AT(17,8):L1$
830 DISPLAY AT(20,1):"SCHREI
BEN":"DATEI :":"ZEILE : >"
840 DISPLAY AT(21,8):L2$
850 LINPUT #1:A$
860 IF A$=E$ THEN 950
870 CALL ZEILE(Z,A$)
880 DISPLAY AT(18,10):Z
890 IF W=2 THEN IF Z<Z1 THEN
 850
900 IF W=1 THEN IF Z>=Z1 AND
 Z<=Z2 THEN 850
910 IF W=2 THEN IF Z>Z2 THEN
 950
920 PRINT #2:A$
930 DISPLAY AT(22,10):Z
940 GOTO 850
950 PRINT #2:E$ :: CLOSE #1
:: CLOSE #2
960 DISPLAY AT(24,1):"FERTIG
! WEITER MIT ENTER"
970 ACCEPT AT(24,26):W$
980 RETURN
990 ! *** LOAD SUB ***
1000 DISPLAY AT(1,1)ERASE AL
L:"LOAD SUB": :"LAEDT PROGRA
MM B  HINTER":"PROGRAMM A":
:"DAS NEUE PROGRAMM WIRD AUT
OMFORTLAUFEND NUMERIERT"
1010 DISPLAY AT(9,1):"SIE KO
ENNEN DIE SCHRITTWEITEFUER P
ROGRAMM B WAEHLEN.....(STAND
ARD IST 10)"
1020 DISPLAY AT(13,1):T1$
1030 DISPLAY AT(19,4)BEEP:">
 1..START": :"...> 2..ZURUEC
K"
1040 CALL WAHL(2,W)
1050 ON W GOTO 1060,380
1060 DISPLAY AT(1,11)ERASE A
LL:"LOAD SUB"
1070 DISPLAY AT(5,1):T2$
1080 DISPLAY AT(8,1):T3$
1090 DISPLAY AT(12,1):T4$
1100 IF I=99 THEN 1150 :: AC
CEPT AT(5,19)SIZE(-10)BEEP:P
1$
1110 IF P1$="" THEN 1100
1120 ACCEPT AT(6,19)SIZE(-10
)BEEP:L1$
1130 IF L1$="" THEN 1120
1140 CALL DATEI(L1$,P1$)
1150 ACCEPT AT(8,19)SIZE(10)
BEEP:P2$
1160 IF P2$="" THEN 1150
1170 ACCEPT AT(9,19)SIZE(10)
BEEP:L2$
1180 IF L2$="" THEN 1170
1190 CALL DATEI(L2$,P2$)
1200 IF L2$=L1$ THEN CALL DA
TFEHLER :: GOTO 1100
1210 ACCEPT AT(10,19)VALIDAT
E(NUMERIC)SIZE(-5)BEEP:S
1220 IF S<1 THEN 1210
1230 IF I=99 THEN 1290
1240 ACCEPT AT(12,19)SIZE(-1
0)BEEP:P3$
1250 IF P3$="" THEN 1240
1260 ACCEPT AT(13,19)SIZE(-1
0)BEEP:L3$
1270 IF L3$="" THEN 1260
1280 CALL DATEI(L3$,P3$)
1290 IF L3$=L1$ OR L3$=L2$ T
HEN CALL DATFEHLER :: GOTO 1
100
1300 IF I<>99 THEN OPEN #1:L
1$,DISPLAY ,VARIABLE 163,INP
UT
1310 OPEN #2:L2$,DISPLAY ,VA
RIABLE 163,INPUT
1320 IF I<99 THEN OPEN #3:L3
$,DISPLAY ,VARIABLE 163,OUTP
UT
1330 DISPLAY AT(16,1):"LESEN
":"DATEI :":"ZEILE : >"
1340 DISPLAY AT(17,8):L1$
1350 DISPLAY AT(20,1):"SCHRE
IBEN":"DATEI :":"ZEILE : >"
1360 DISPLAY AT(21,8):L3$
1370 IF I=99 THEN 1460
1380 LINPUT #1:A$
1390 IF A$=E$ THEN 1450
1400 CALL ZEILE(Z,A$)
1410 DISPLAY AT(18,10):Z
1420 PRINT #3:A$
1430 DISPLAY AT(22,10):Z
```

```
1440 GOTO 1380
1450 CLOSE #1
1460 Z2=Z
1470 DISPLAY AT(17,1):"SUCHL
AUF SPRUNGADRESSEN"
1480 OPEN #1:"DSK1.REN",INTE
RNAL,VARIABLE
1490 LINPUT #2:A$
1500 IF A$=E$ THEN 1550
1510 CALL ZEILE(Z,A$)
1520 PRINT #1:Z
1530 DISPLAY AT(18,10):Z
1540 GOTO 1490
1550 PRINT #1:65535 :: RESTO
RE #2
1560 DISPLAY AT(17,1):"DATEI
 :"
1570 DISPLAY AT(17,8)BEEP:L2
$
1580 Z=Z2 :: N=0
1590 LINPUT #2:A$
1600 N=N+1
1610 IF A$=E$ THEN 1710
1620 CALL ZEILE(Z1,A$)
1630 DISPLAY AT(18,10):Z1
1640 RESTORE #1
1650 CALL REN(A$,Z2,S)
1660 Z=Z+S
1670 CALL NEUZEILE(Z,A$)
1680 DISPLAY AT(22,10):Z
1690 PRINT #3:A$
1700 GOTO 1590
1710 CLOSE #2
1720 CLOSE #1:DELETE
1730 DISPLAY AT(24,1)BEEP:"W
EITER ANFUEGEN ? (J/N)"
1740 CALL KEY(2,R,S):: IF S=
0 THEN 1740
1750 IF R=2 THEN I=99 :: DIS
PLAY AT(24,1):RPT$(" ",30)::
 GOTO 1150
1760 IF R<>15 THEN 1740
1770 PRINT #3:E$ :: CLOSE #3
1780 RETURN
1790 SUB DATEI(L$,P$)
1800 IF L$="1" OR L$="2" OR
L$="3" THEN 1820
1810 L$="DSK."&L$&"."&P$ ::
GOTO 1830
1820 L$="DSK"&L$&"."&P$
1830 SUBEND
1840 SUB DATFEHLER
1850 DISPLAY AT(20,5):"UNERL
AUBTE DATEIKOMBINATION.....
.....WEITE
R MIT ENTER"
1860 CALL SOUND(200,220,0)
1870 ACCEPT AT(24,1):W$
1880 DISPLAY AT(20,1):RPT$("
 ",96)
1890 SUBEND
1900 SUB WAHL(I,W)
1910 CALL KEY(5,R,S)
1920 IF S=0 THEN 1910
1930 W=R-48
1940 IF W<1 OR W>I THEN 1910
1950 SUBEND
1960 SUB ZEILE(Z,A$)
1970 Z$=SEG$(A$,1,2)
1980 CALL DEZ(Z,Z$)
1990 SUBEND
2000 SUB DEZ(Z,Z$)
2010 Z=ASC(Z$)*256+ASC(SEG$(
Z$,2,1))
2020 SUBEND
2030 SUB NEUZEILE(Z,A$)
2040 Z$=CHR$(INT(Z/256))&CHR
$(Z-INT(Z/256)*256)
2050 A$=Z$&SEG$(A$,3,LEN(A$)
-2)
2060 SUBEND
2070 SUB REN(A$,Z1,S)
2080 FOR I=3 TO LEN(A$)-3
2090 Z$=SEG$(A$,I,1)
2100 IF Z$=CHR$(131)OR Z$=CH
R$(154)THEN 2260
2110 IF Z$<>CHR$(201)THEN 22
50
2120 I=I+1
2130 B$=SEG$(A$,I,2)
2140 CALL ZEILE(Z,B$)
2150 N=1
2160 RESTORE #1
2170 INPUT #1:Z2,
2180 IF EOF(1)<>0 THEN Z=327
67 :: GOTO 2220
2190 IF Z2=Z THEN 2210
2200 N=N+1 :: GOTO 2170
2210 Z=Z1+N*S
2220 Z$=CHR$(INT(Z/256))&CHR
$(Z-INT(Z/256)*256)
2230 A$=SEG$(A$,1,I-1)&Z$&SE
G$(A$,I+2,LEN(A$)-I-1)
2240 I=I+2
2250 NEXT I
2260 SUBEND
2270 SUB PH(N$,D$)
2280 CALL CLEAR
2290 CALL CHAR(128,"3F3F3030
303F3F303030303F3F303030F0F8
1C0C1CF8F0000C0C0CFCFC0C0C0C
")
```

```
2300 CALL MAGNIFY(4)
2310 CALL SPRITE(#1,128,5,8,
120)
2320 DISPLAY AT(8,12):"PROGR
AMM"
2330 L=LEN(N$)+4
2340 S=16-L/2
2350 DISPLAY AT(11,S):RPT$("
*",L)
2360 DISPLAY AT(12,S):"*";RP
T$(" ",L-2);"*"
2370 DISPLAY AT(13,S):"* ";N
$;" *"
2380 DISPLAY AT(14,S):"*";RP
T$(" ",L-2);"*"
2390 DISPLAY AT(15,S):RPT$("
*",L)
2400 DISPLAY AT(17,S+6):D$
2410 DISPLAY AT(20,11):"COPY
RIGHT":TAB(8);"PETER HEIMLIC
H"
2420 INPUT "WEITER MIT ENTER
":W$
2430 CALL DELSPRITE(ALL)
2440 SUBEND
```

```
100 ! ************************
110 ! *  GESAMTER SPRACH-   *
120 ! * SCHATZ DES SPEECH-  *
130 ! *    SYNTHESIZERS     *
140 ! *   COPYRIGHT 1985    *
150 ! * by Werner Liebener  *
160 ! *   BEARBEITET VOM    *
170 ! *       TEAM DES      *
180 ! *  AKTUELL VERLAGES   *
190 ! *       MUENCHEN      *
200 ! * BENOETIGTE GERAETE  *
210 ! *  TI 99/4A KONSOLE   *
220 ! * SPEECH-SYNTHESIZER  *
230 ! *  EXTENDED - BASIC   *
240 ! * KASS.-REC. od. DSK  *
250 ! *                     *
260 ! *    BENOETIGTER      *
270 ! *   SPEICHERPLATZ     *
280 ! *    8054  BYTES      *
290 ! *                     *
300 ! ************************

310 CALL CLEAR
320 CALL SCREEN(1)
330 PRINT ".......WORTSCHATZ
 DES..":
340 PRINT " SPRACHSYNTHESIZE
RS PHP 1500 ":
350 PRINT ".....FUER DEN TI
99/4A": :
360 PRINT "...**************
********..."
370 PRINT "...*....PROGRAMM
VON....*......*.............
.......*"
380 PRINT "...*..WERNER LIEB
ENDER..*......*.............
.......*."
390 PRINT "...*.BISMARCKSTRA
SSE 21 *......*.............
.......*"
400 PRINT "...*...2000 HAMBU
RG 19..*..."
410 PRINT "...**************
********...": :
420 PRINT "..>.0.<.*ZAHLEN":
 :
430 PRINT "..>.§.<.*ALPHABET
": :
440 PRINT "..>A-Z<.*WORTE":
:
450 PRINT "..>.=.<.*GESAMTER
.WORTSCHATZ"
460 CALL SCREEN(15)
470 CALL KEY(0,K,S)
480 IF S=0 THEN 470
```

```
490 IF K=48 THEN 790
500 IF K-48<12 THEN 470
510 IF K=64 THEN 870
520 IF K=65 THEN 950
530 IF K=66 THEN 1030
540 IF K=67 THEN 1110
550 IF K=68 THEN 1190
560 IF K=69 THEN 1270
570 IF K=70 THEN 1350
580 IF K=71 THEN 1430
590 IF K=72 THEN 1510
600 IF K=73 THEN 1590
610 IF K=74 THEN 1670
620 IF K=75 THEN 1750
630 IF K=76 THEN 1830
640 IF K=77 THEN 1910
650 IF K=78 THEN 1990
660 IF K=79 THEN 2070
670 IF K=80 THEN 2150
680 IF K=81 THEN 2230
690 IF K=82 THEN 2310
700 IF K=83 THEN 2390
710 IF K=84 THEN 2470
720 IF K=85 THEN 2550
730 IF K=86 THEN 2630
740 IF K=87 THEN 2710
750 IF K=88 THEN 2790
760 IF K=89 THEN 2870
770 IF K=90 THEN 2950
780 IF K=61 THEN 3410
790 RESTORE 3040
800 CALL CLEAR
810 FOR I=1 TO 25
820 READ ZAHL$
830 PRINT TAB(10);ZAHL$
840 CALL SAY(ZAHL$)
850 NEXT I
860 GOTO 310
870 RESTORE 3060
880 CALL CLEAR
890 FOR I=1 TO 26
900 READ §$
910 PRINT TAB(10);§$
920 CALL SAY(§$)
930 NEXT I
940 GOTO 310
950 RESTORE 3070
960 CALL CLEAR
970 FOR I=1 TO 15
980 READ A$
990 PRINT TAB(10);A$
1000 CALL SAY(A$)
1010 NEXT I
1020 GOTO 310
1030 RESTORE 3080
1040 CALL CLEAR
1050 FOR I=1 TO 13
1060 READ B$
1070 PRINT TAB(10);B$
1080 CALL SAY(B$)
1090 NEXT I
1100 GOTO 310
1110 RESTORE 3090
1120 CALL CLEAR
1130 FOR I=1 TO 20
1140 READ C$
1150 PRINT TAB(10);C$
1160 CALL SAY(C$)
1170 NEXT I
1180 GOTO 310
1190 RESTORE 3110
1200 CALL CLEAR
1210 FOR I=1 TO 15
1220 READ D$
1230 PRINT TAB(10);D$
1240 CALL SAY(D$)
1250 NEXT I
1260 GOTO 310
1270 RESTORE 3120
1280 CALL CLEAR
1290 FOR I=1 TO 12
1300 READ E$
1310 PRINT TAB(10);E$
1320 CALL SAY(E$)
1330 NEXT I
1340 GOTO 310
1350 RESTORE 3130
1360 CALL CLEAR
1370 FOR I=1 TO 18
1380 READ F$
1390 PRINT TAB(10);F$
1400 CALL SAY(F$)
1410 NEXT I
1420 GOTO 310
1430 RESTORE 3150
1440 CALL CLEAR
1450 FOR I=1 TO 16
1460 READ G$
1470 PRINT TAB(10);G$
1480 CALL SAY(G$)
1490 NEXT I
1500 GOTO 310
1510 RESTORE 3160
1520 CALL CLEAR
1530 FOR I=1 TO 17
1540 READ H$
1550 PRINT TAB(10);H$
1560 CALL SAY(H$)
1570 NEXT I
1580 GOTO 310
```

```
1590 RESTORE 3170
1600 CALL CLEAR
1610 FOR I=1 TO 10
1620 READ I$
1630 PRINT TAB(10);I$
1640 CALL SAY(I$)
1650 NEXT I
1660 GOTO 310
1670 RESTORE 3180
1680 CALL CLEAR
1690 FOR I=1 TO 3
1700 READ J$
1710 PRINT TAB(10);J$
1720 CALL SAY(J$)
1730 NEXT I
1740 GOTO 310
1750 RESTORE 3190
1760 CALL CLEAR
1770 FOR I=1 TO 4
1780 READ K$
1790 PRINT TAB(10);K$
1800 CALL SAY(K$)
1810 NEXT I
1820 GOTO 310
1830 RESTORE 3200
1840 CALL CLEAR
1850 FOR I=1 TO 17
1860 READ L$
1870 PRINT TAB(10);L$
1880 CALL SAY(L$)
1890 NEXT I
1900 GOTO 310
1910 RESTORE 3210
1920 CALL CLEAR
1930 FOR I=1 TO 16
1940 READ M$
1950 PRINT TAB(10);M$
1960 CALL SAY(M$)
1970 NEXT I
1980 GOTO 310
1990 RESTORE 3220
2000 CALL CLEAR
2010 FOR I=1 TO 13
2020 READ N$
2030 PRINT TAB(10);N$
2040 CALL SAY(N$)
2050 NEXT I
2060 GOTO 310
2070 RESTORE 3230
2080 CALL CLEAR
2090 FOR I=1 TO 12
2100 READ O$
2110 PRINT TAB(10);O$
2120 CALL SAY(O$)
2130 NEXT I
2140 GOTO 310
2150 RESTORE 3240
2160 CALL CLEAR
2170 FOR I=1 TO 19
2180 READ P$
2190 PRINT TAB(10);P$
2200 CALL SAY(P$)
2210 NEXT I
2220 GOTO 310
2230 RESTORE 3260
2240 CALL CLEAR
2250 FOR I=1 TO 1
2260 READ Q$
2270 PRINT TAB(10);Q$
2280 CALL SAY(Q$)
2290 NEXT I
2300 GOTO 310
2310 RESTORE 3270
2320 CALL CLEAR
2330 FOR I=1 TO 13
2340 READ R$
2350 PRINT TAB(10);R$
2360 CALL SAY(R$)
2370 NEXT I
2380 GOTO 310
2390 RESTORE 3280
2400 CALL CLEAR
2410 FOR I=1 TO 39
2420 READ S$
2430 PRINT TAB(10);S$
2440 CALL SAY(S$)
2450 NEXT I
2460 GOTO 310
2470 RESTORE 3310
2480 CALL CLEAR
2490 FOR I=1 TO 40
2500 READ T$
2510 PRINT TAB(10);T$
2520 CALL SAY(T$)
2530 NEXT I
2540 GOTO 310
2550 RESTORE 3340
2560 CALL CLEAR
2570 FOR I=1 TO 8
2580 READ U$
2590 PRINT TAB(10);U$
2600 CALL SAY(U$)
2610 NEXT I
2620 GOTO 310
2630 RESTORE 3350
2640 CALL CLEAR
2650 FOR I=1 TO 3
2660 READ V$
2670 PRINT TAB(10);V$
2680 CALL SAY(V$)
```

```
2690 NEXT I
2700 GOTO 310
2710 RESTORE 3360
2720 CALL CLEAR
2730 FOR I=1 TO 26
2740 READ W$
2750 PRINT TAB(10);W$
2760 CALL SAY(W$)
2770 NEXT I
2780 GOTO 310
2790 RESTORE 3380
2800 CALL CLEAR
2810 FOR I=1 TO 1
2820 READ X$
2830 PRINT TAB(10);X$
2840 CALL SAY(X$)
2850 NEXT I
2860 GOTO 310
2870 RESTORE 3390
2880 CALL CLEAR
2890 FOR I=1 TO 7
2900 READ Y$
2910 PRINT TAB(10);Y$
2920 CALL SAY(Y$)
2930 NEXT I
2940 GOTO 310
2950 RESTORE 3400
2960 CALL CLEAR
2970 FOR I=1 TO 2
2980 READ Z$
2990 PRINT TAB(10);Z$
3000 CALL SAY(Z$)
3010 NEXT I
3020 GOTO 310
3030 DATA THIS+ARE+ALL+WORDS
+I+HAVE
3040 DATA -1,+1,0.1,0,1,2,3,
4,5,6,7,8,9,TEN,ELEVEN,TWELV
E,THIRTEEN,FOURTEEN,FIFTEEN,
SIX+TEEN,SEVEN+TEEN
3050 DATA EIGHT+TEEN,NINE+TE
EN,TWENTY,IT+GOES+TO+NINE+HU
NDRED+NINETY+NINE
3060 DATA A,B,C,D,E,F,G,H,I,
J,K,L,M,N,O,P,Q,R,S,T,U,V,W,
X,Y,Z
3070 DATA A,A1,ABOUT,AFTER,A
GAIN,ALL,AM,AN,AND,ANSWER,AN
Y,ARE,AS,ASSUME,AT
3080 DATA B,BACK,BASE,BE,BET
WEEN,BLACK,BLUE,BOTH,BOTTOM,
BUT,BUY,BY,BYE
3090 DATA C,CAN,CASSETTE,CEN
TER,CHECK,CHOICE,CLEAR,COLOR
,COME,COMES,COMMA,COMMAND,CO
MPLETE,COMPLETED,COMPUTER
3100 DATA CONNECTED,CONSOLE,
CORRECT,COURSE,CYAN
3110 DATA D,DATA,DECIDE,DEVI
CE,DID,DIFFERENT,DISKETTE,DO
,DOES,DOING,DONE,DOUBLE,DOWN
,DRAW,DRAWING
3120 DATA E,EACH,EIGHT,EIGHT
Y,ELEVEN,ELSE,END,ENDS,ENTER
,ERROR,EXACTLY,EYE
3130 DATA F,FIFTEEN,FIFTY,FI
GURE,FIND,FINE,FINISH,FINISH
ED,FIRST,FIT,FIVE,FOR,FORTY,
FOUR,FOURTEEN,FOURTH,FROM
3140 DATA FRONT
3150 DATA G,GAMES,GET,GETTIN
G,GIVE,GIVES,GO,GOES,GOING,G
OOD,GOOD WORK,GOODBYE,GOT,GR
AY,GREEN,GUESS
3160 DATA H,HAD,HAND,#HANDHE
LD UNIT#,HAS,HAVE,HEAD,HEAR,
HELLO,HELP,HERE,HIGHER,HIT,H
OME,HOW,HUNDRED,HURRY
3170 DATA I,#I WIN#,IF,IN,IN
CH,INCHES,INSTRUCTION,INSTRU
CTIONS,IS,IT
3180 DATA J,JOYSTICK,JUST
3190 DATA K,KEY,KEYBOARD,KNO
W
3200 DATA L,LARGE,LARGER,LAR
GEST,LAST,LEARN,LEFT,LESS,LE
T,LIKE,LIKES,LINE,LOAD,LONG,
LOOK,LOOKS,LOWER
3210 DATA M,MADE,MAGENTA,MAK
E,ME,MEAN,MEMORY,MESSAGE,MES
SAGES,MIDDLE,MIGHT,MODULE,MO
RE,MOST,MOVE,MUST
3220 DATA N,NAME,NEAR,NEED,N
EGATIVE,NEXT,#NICE TRY#,NINE
,NINETY,NO,NOT,NOW,NUMBER
3230 DATA O,OF,OFF,OH,ON,ONE
,ONLY,OR,ORDER,OTHER,OUT,OVER
3240 DATA P,PART,PARTNER,PAR
TS,PERIOD,PLAY,PLAYS,PLEASE,
POINT,POSITION,POSITIVE,PRES
S,PRINT,PRINTER,PROBLEM
3250 DATA PROBLEMS,PROGRAM,P
UT,PUTTING
3260 DATA Q
3270 DATA R,RANDOMLY,READ,RE
AD1,#READY TO START#,RECORDE
R,RED,REFER,REMEMBER,RETURN,
REWIND,RIGHT,ROUND
3280 DATA S,SAID,SAVE,SAY,SA
YS,SCREEN,SECOND,SEE,SEES,SE
T,SEVEN,SEVENTY,SHAPE,SHAPES
,SHIFT,SHORT,SHORTER,SHOULD
```

```
3290 DATA SIDE,SIDES,SIX,SIX
TY,SMALL,SMALLER,SMALLEST,SO
,SOME,SORRY,SPACE,SPACES,SPE
LL,SQUARE,START,STEP,STOP
3300 DATA SUM,SUPPOSED,#SUPP
OSED TO#,SURE
3310 DATA T,TAKE,TEEN,TELL,T
EN,#TEXAS INSTRUMENTS#,THAN,
THAT,#THAT IS INCORRECT#,#TH
AT IS RIGHT#,THE,THE1,THEIR
3320 DATA THEN,THERE,THESE,T
HEY,THING,THINGS,THINK,THIRD
,THIRTEEN,THIRTY,THIS,THREE,
THREW,THROUGH,TIME,TO
3330 DATA TOGETHER,TONE,TOO,
TOP,TRY,#TRY AGAIN#,TURN,TWE
LVE,TWENTY,TWO,TYPE
3340 DATA U,UHOH,UNDER,UNDER
STAND,UNTIL,UP,UPPER,USE
3350 DATA V,VARY,VERY
3360 DATA W,WAIT,WANT,WANTS,
WAY,WE,WEIGH,WEIGHT,WELL,WER
E,WHAT,#WHAT WAS THAT#,WHEN,
WHERE,WHICH,WHITE,WHO,WHY,WI
LL
3370 DATA WITH,WON,WORD,WORD
S,WORK,WORKING,WRITE
3380 DATA X
3390 DATA Y,YELLOW,YES,YET,Y
OU,#YOU WIN#,YOUR
3400 DATA Z,ZERO
3410 RESTORE 3030
3420 CALL CLEAR
3430 FOR I=1 TO 412
3440 READ ALLE$
3450 PRINT TAB(10);ALLE$
3460 CALL SAY(ALLE$)
3470 NEXT I
3480 GOTO 310
```

```
100 ! **********************
110 ! *   PIXEL - MASTER    *
120 ! *   COPYRIGHT 1985    *
130 ! * by Sascha  Nungess *
140 ! *   BEARBEITET VOM    *
150 ! *       TEAM DES      *
160 ! *  AKTUELL VERLAGES   *
170 ! *       MUENCHEN      *
180 ! * BENOETIGTE GERAETE *
190 ! *  TI 99/4A KONSOLE   *
200 ! * KASS.REC. oder DSK *
210 ! *   EXTENDED BASIC    *
220 ! *                     *
230 ! *    BENOETIGTER      *
240 ! *   SPEICHERPLATZ     *
250 ! *    6334   BYTES     *
260 ! *                     *
270 ! **********************
280 CALL CLEAR
290 CALL SCREEN(2)
300 FOR I=0 TO 14 :: CALL CO
LOR(I,15,1):: NEXT I
310 DISPLAY AT(8,3):"P I X E
 L - M A S T E R" :: DISPLAY
 AT(24,14):"(C) BY PIRANHA"
320 T=400
330 CALL SOUND(T,330,2,131,6
)
340 CALL SOUND(T,330,3,131,6
)
350 CALL SOUND(T,349,3,131,6
)
360 CALL SOUND(T,392,2,131,6
)
370 CALL SOUND(T,392,3,147,6
)
380 CALL SOUND(T,349,3,147,6
)
390 CALL SOUND(T,330,3,147,6
)
400 CALL SOUND(T,294,3,147,6
)
410 CALL SOUND(T,262,2,165,6
)
420 CALL SOUND(T,262,3,165,7
)
430 CALL SOUND(T,294,2,175,6
)
440 CALL SOUND(T,330,2,176,5
)
450 CALL SOUND(T*1.5,294,2,1
96,7)
460 CALL SOUND(T/2,262,3,196
,7)
470 CALL SOUND(2*T,262,4,165
,7,131,8)
480 CALL CLEAR
```

```
490 DISPLAY AT(1,4):"MENUE E
RKLAERUNG :"
500 DISPLAY AT(3,1):"1 : KRE
IS(ELLYPSE)" :: DISPLAY AT(5
,1):"2 : BOGEN" :: DISPLAY A
T(7,1):"3 : LINIE" :: DISPLA
Y AT(9,1):"4 : GERADE"
510 DISPLAY AT(11,1):"5 : RE
CHTECK" :: DISPLAY AT(13,1):
"6 : PIXEL" :: DISPLAY AT(15
,1):"7 : DEMO PROGRAMM"
520 DISPLAY AT(17,1):"8 : BI
LDSCHIRM LOESCHEN" :: DISPLA
Y AT(19,1):"9 : PROGRAMM ERK
LAERUNG"
530 DISPLAY AT(24,4):"TASTE
DRUECKEN"
540 CALL KEY(0,K,S):: IF S=0
 THEN 540
550 CALL CLEAR
560 DISPLAY AT(1,6):"VARIABL
EN :" :: DISPLAY AT(3,2):"A/
E : ANFANGS UND ENDWINKEL" :
: DISPLAY AT(5,4):"L : LAENG
E"
570 DISPLAY AT(7,4):"M : LAE
NGE KANTE A" :: DISPLAY AT(9
,4):"N : LAENGE KANTE B" ::
DISPLAY AT(11,4):"R : SENKRE
CHTER RADIUS"
580 DISPLAY AT(13,3):"R1 : W
AAGRECHTER RADIUS" :: DISPLA
Y AT(15,2):"X/Y : SPALTE / Z
EILE" :: DISPLAY AT(17,1):"X
1/Y1: SPALTE/ZEILE 1.PUNKT"
590 DISPLAY AT(19,1):"X2/Y2:
 SPALTE/ZEILE 2.PUNKT"
600 DISPLAY AT(24,4):"TASTE
DRUECKEN"
610 CALL KEY(0,K,S):: IF S=0
 THEN 610
620 CALL CLEAR
630 !--------MENUE--------!
640 DISPLAY AT(1,1):"1 KR."
:: DISPLAY AT(3,1):"2 BO." :
: DISPLAY AT(5,1):"3 LI9" ::
 DISPLAY AT(7,1):"4 GE."
650 DISPLAY AT(9,1):"5 RE."
:: DISPLAY AT(11,1):"6 PI."
:: DISPLAY AT(13,1):"7 DE."
:: DISPLAY AT(15,1):"8 CL."
:: DISPLAY AT(17,1):"9 PE."
660 CALL KEY(0,K,S):: DISPLA
Y AT(24,1):"................
.........." :: IF S=0 THEN 6
60 :: IF K<49 OR K>57 THEN 6
60
670 ON K-48 GOTO 740,780,850
,900,950,1010,1060,690,480
680 !-BILDSCHIRM LOESCHEN-!
690 DISPLAY AT(24,1):"BIST D
U DIR SICHER (J/N)"
700 CALL KEY(0,K,S):: IF S=0
 THEN 700 :: IF K=74 THEN 72
0 :: IF K=78 THEN 710
710 DISPLAY AT(24,1):"......
.................." :: GOT
O 660
720 GOTO 620
730 !----ABFRAGE KREIS----!
740 DISPLAY AT(24,1):"X=....
Y=....R=....R1=" :: ACCEPT A
T(24,3)SIZE(3):X :: ACCEPT A
T(24,9)SIZE(3):Y
750 ACCEPT AT(24,15)SIZE(3):
R :: ACCEPT AT(24,22)SIZE(3)
:R1
760 CALL KREIS(Y,X,R,R1)
770 GOTO 660
780 !----ABFRAGE BOGEN----!
790 DISPLAY AT(24,4):"X=....
Y=....R="
800 ACCEPT AT(24,6)SIZE(3):X
 :: ACCEPT AT(24,12)SIZE(3):
Y :: ACCEPT AT(24,18)SIZE(3)
:R
810 DISPLAY AT(24,4):"R1=...
.A=.....E="
820 ACCEPT AT(24,7)SIZE(3):R
1 :: ACCEPT AT(24,13)SIZE(3)
:A :: ACCEPT AT(24,20)SIZE(3
):E
830 CALL BOGEN(Y,X,R,R1,A,E)
840 GOTO 660
850 !----ABFRAGE LINIE----!
860 DISPLAY AT(24,1):"X=....
Y=....A=.....L="
870 ACCEPT AT(24,3)SIZE(3):X
 :: ACCEPT AT(24,9)SIZE(3):Y
 :: ACCEPT AT(24,15)SIZE(3):
A :: ACCEPT AT(24,22)SIZE(3)
:L
880 CALL LINIE(Y,X,A,L)
890 GOTO 660
900 !---ABFRAGE GERADE---!
910 DISPLAY AT(24,1):"X1=...
.Y1=....X2=....Y2="
920 ACCEPT AT(24,4)SIZE(3):X
1 :: ACCEPT AT(24,11)SIZE(3)
:Y1 :: ACCEPT AT(24,18)SIZE(
3):X2 :: ACCEPT AT(24,25)SIZ
E(3):Y2
```

```
930 CALL GERADE(Y1,X1,Y2,X2)
940 GOTO 660
950 !--ABFRAGE RECHTECK--!
960 DISPLAY AT(24,1):"X=....
Y=....M=...N=...A="
970 ACCEPT AT(24,3)SIZE(3):X
 :: ACCEPT AT(24,9)SIZE(3):Y
 :: ACCEPT AT(24,15)SIZE(2):
M
980 ACCEPT AT(24,20)SIZE(2):
N :: ACCEPT AT(24,25)SIZE(3)
:A
990 CALL RECHTECK(Y,X,M,N,A)
1000 GOTO 660
1010 !----ABFRAGE PIXEL----!
1020 DISPLAY AT(24,6):"X1=..
....Y1="
1030 ACCEPT AT(24,9)SIZE(4):
X1 :: ACCEPT AT(24,18)SIZE(3
):Y1
1040 CALL PLOT(Y1,X1)
1050 GOTO 660
1060 !-----DEMO-----!
1070 CALL CLEAR
1080 CALL RECHTECK(20,-5,40,
15,0)
1090 CALL RECHTECK(30,35,15,
30,0)
1100 CALL GERADE(5,-5,-5,-9)
1110 CALL RECHTECK(28,40,5,5
,0)
1120 CALL RECHTECK(30,5,3,10
,0)
1130 CALL RECHTECK(25,20,5,5
,0)
1140 CALL KREIS(0,0,5,5)
1150 CALL KREIS(0,15,5,5)
1160 CALL KREIS(0,30,5,5)
1170 CALL GERADE(0,0,0,30)
1180 DISPLAY AT(24,3):"BITTE
 TASTE DRUECKEN"
1190 CALL KEY(0,K,S):: IF S=
0 THEN 1190
1200 GOTO 620
1210 !-----LINIE-----!
1220 SUB LINIE(Y,X,A,L)
1230 Q=(A)*(PI/180)
1240 V=SIN(Q)
1250 W=COS(Q)
1260 FOR I=1 TO L
1270 CALL PLOT(Y,X)
1280 Y=Y+V
1290 X=X+W
1300 NEXT I
1310 SUBEND
1320 !-----GERADE-----!
1330 SUB GERADE(Y1,X1,Y2,X2)
1340 YD=Y2-Y1
1350 XD=X2-X1
1360 SW=MAX(ABS(YD),ABS(XD))
1370 SY=YD/SW
1380 SX=XD/SW
1390 FOR I=1 TO SW
1400 CALL PLOT(Y1,X1)
1410 Y1=Y1+SY
1420 X1=X1+SX
1430 NEXT I
1440 SUBEND
1450 !-----RECHTECK-----!
1460 SUB RECHTECK(Y,X,M,N,A)
1470 CALL LINIE(Y,X,A,M)
1480 CALL LINIE(Y,X,A+270,N)
1490 CALL LINIE(Y,X,A+180,M)
1500 CALL LINIE(Y,X,A+90,N)
1510 SUBEND
1520 !-----BOGEN-----!
1530 SUB BOGEN(Y,X,R,R1,A,E)
1540 A=A*(PI/180)
1550 E=E*(PI/180)
1560 FOR AL=A TO E STEP 1/R
1570 Y1=INT(R*SIN(AL)+.5)+Y
1580 X1=INT(R1*COS(AL)+.5)+X
1590 CALL PLOT(Y1,X1)
1600 NEXT AL
1610 SUBEND
1620 !-----KREIS-----!
1630 SUB KREIS(Y,X,R,R1)
1640 CALL BOGEN(Y,X,R,R1,0,3
59)
1650 SUBEND
1660 !-----PIXEL-----!
1670 SUB PLOT(Y1,X1)
1680 ! WERTE UEBERPRUEFEN
1690 IF Y1<-95 THEN 2140
1700 IF X1<-127 THEN 2140
1710 IF Y1>96 THEN 2140
1720 IF X1>128 THEN 2140
1730 ! OFFSET BERECHNEN
1740 Y=INT(Y1-1)*(-1)+95
1750 X=INT(X1)+127
1760 ! TEST OB 1.DURCHLAUF
1770 IF F=0 THEN 2070
1780 !----BERECHNUNGEN----!
1790 XS=INT(X/8)+1
1800 YS=INT(Y/8)+1
1810 XP=8*((X/8)-INT(X/8))
1820 YP=8*((Y/8)-INT(Y/8))
1830 XH=(YP*2+1)+INT(XP/4)
1840 XB=3-(XP-INT(XP/4)*4)
1850 ! TEST NEUES ZEICHEN
```

```
1860 CALL GCHAR(YS,XS,Z)
1870 IF Z>90 THEN 1930
1880 ZE=ZE-1
1890 Z=ZE
1900 !TEST LETZTES ZEICHEN
1910 IF ZE<91 THEN 2140
1920 !  STRING ZERLEGEN
1930 CALL CHARPAT(Z,H$)
1940 G$=SEG$(H$,XH,1)
1950 G=POS(HX$,G$,1)-1
1960 Q=(2^XB)OR G
1970 G$=SEG$(HX$,Q+1,1)
1980 ! STRING ADDIEREN
1990 B$=SEG$(H$,1,XH-1)
2000 C$=SEG$(H$,XH+1,16-XH)
2010 H$=B$&G$&C$
2020 ! ZEICHEN DEF.,SETZTEN
2030 CALL CHAR(Z,H$)
2040 CALL HCHAR(YS,XS,Z)
2050 GOTO 2140
2060 !    STARTROUTINE
2070 ZE=143 :: F=1
2080 HX$="0123456789ABCDEF"
2090 FOR I=91 TO 143
2100 CALL CHAR(I,"0")
2110 NEXT I
2120 GOTO 1790
2130 !  ENDE DER ROUTINE
2140 SUBEND
```

FUNKTIONSPLOT

Dieses Programm stellt mathematische Funktionen in Einzelpunktauflösung dar. In Zeile 330 ist die gewünschte Funktion vor Programmstart einzugeben. Die Sinusfunktion in dieser Zeile dient nur als Demonstrationsbeispiel und hat Platzhalterfunktionen für Ihre eigenen Formeln. Nach dem Programmstart geben Sie die untere und obere Grenze der X-Achse sowie die Schrittweite der Pixels ein. Als Schrittgröße wird ein Wert zwischen 0.5 und 2 empfohlen, da die Auflösung hier am optimalsten ist. Hat der Rechner seine Darstellung beendet, macht er sich mit einem Signalton bemerkbar. Das Programm kann dann durch Drücken einer beliebigen Taste abgebrochen werden.

Andreas Braun

STERNENHIMMEL

Mit diesem Programm kann der Sternenhimmel an jeder beliebigen Stelle der Erde dargestellt werden.
Geben Sie den Breitekreis, Längengrad, das Datum und die Uhrzeit ein. Danach wird der Computer die Koordinaten der Sternbilder über den angegebenen Schnittpunkten berechnen und die Sterne in Einzelpunktauflösung auf dem Monitor darstellen. Ist der letzte Stern generiert, macht Sie der Rechner mit einem Signalton darauf aufmerksam, daß er seine Arbeit beendet hat. Mit der Enter-Taste können Sie nun neue Koordinaten eingeben.

Udo Wenzel

KLANG

Dieses Programm ermöglicht auch dem unmusikalischsten unserer Leser, die passenden Geräusche oder musikalischen Effekte für sein Programm zu finden.
HANDHABUNG:
1. Frequenzen
Die Frequenzen der drei Tongeneratoren lassen sich mit den Tasten 1, 2, 3 jeweils um den Wert „Frequenz-Additiv" erhöhen. Werden die Tasten 1, 2 und 3 zusammen mit der Shifttaste gedrückt, so erniedrigen sich die Werte. Wird der mögliche Frequenzbereich über- oder unterschritten, so stellt sich automatisch der Wert 110 ein. Der Rauschgenerator wird mit Taste 4 jeweils um 1 erniedrigt und mit SHIFT –4 um 1 erhöht. Der bei Überschreitung der Grenzen eingestellte Wert beträgt –4.
2. Lautstärke
Die Lautstärke wird mit der Taste –L– erhöht (die Zahl wird geringer) und mit –SHIFT L– verringert (die Zahl wird höher). Der Fehlerwert ist hier die Zahl 8.
3. Tondauer
Die Tondauer kann ebenfalls mit dem Wert „Frequenz-Additiv" durch Drücken von –SHIFT D– verändert werden. Das Vorzeichen wird mit der Taste –v– geändert.
4. Additive
Die Additive werden durch Drücken des davor stehenden Buchstabens gewählt.
Der Abbruch des Programmes erfolgt mit –SHIFT E–.
Die Groß- und Kleinschreibung ist unbedingt einzuhalten. Sämtliche durch oben beschriebene Routine erzeugten Werte werden auf dem Bildschirm angezeigt und Ihr könnt sie in Eure Programme miteinbauen. Auf Eure nächsten eingesendeten Programme sind wir jetzt schon gespannt.

```
100 REM**********************
110 REM*                    *
120 REM*  FUNKTIONSPLOT     *
130 REM*  Copyright  by     *
140 REM*  Andreas Braun     *
150 REM*                    *
160 REM* Bearbeitet  vom    *
170 REM*     Team  des      *
180 REM* Aktuell Verlages   *
190 REM*      Muenchen      *
200 REM*                    *
210 REM*Benoetigte Geraete* 
220 REM* TI99/4A  Konsole   *
230 REM*                    *
270 REM* Speicherbelegung   *
280 REM*     1447 Bytes     *
290 REM**********************
295 REM
300 CALL CLEAR
310 PRINT TAB(9);"FUNKTIONSP
LOT":TAB(9);"============":
 : :
320 GOTO 350
330 Y=SIN(X)*X
340 RETURN
350 INPUT "UNTERE X-GRENZE:
":U
360 PRINT
370 INPUT "OBERE..X-GRENZE:
":O
380 PRINT
390 INPUT "AUFLOESUNG zB:1:
":A
400 PRINT : : :
410 PRINT "X-STRECKUNG WIRD
BERECHNET !": :
420 DI=ABS(O-U)
430 ST=DI/224
440 X=U
450 PRINT "Y-STRECKUNG WIRD
BERECHNET !": : :
460 GOSUB 330
470 MAX=Y
480 MIN=Y
490 FOR X=U+ST TO O STEP ST*
A
500 GOSUB 330
510 IF Y>MIN THEN 530
520 MIN=Y
530 IF Y<MAX THEN 550
540 MAX=Y
550 NEXT X
560 YDI=ABS(MAX-MIN)
570 PRINT "VARIABELN WERDEN
FESTGELEGT!":
580 DIM A$(127)
590 FOR I=0 TO 127
600 A$(I)="0"
610 NEXT I
620 ST$="84219531A623B733C46
5D575E667F7778CA99DB9AEABBFB
BCCEDDDFDEEEFFFFF"
630 CR=31
640 CALL HCHAR(1,1,31,768).
650 X=U-ST*A
660 FOR E=17 TO 241 STEP A
670 X=X+ST*A
680 GOSUB 330
690 Y=4-Y*184/YDI+MAX*184/YD
I
700 IF (Y<1)+(Y>192)THEN 720
710 GOSUB 740
720 NEXT E
730 GOTO 940
740 YI=INT(Y)
750 XI=INT(E)
760 YC=INT(YI/8)
770 XC=INT(XI/8)
780 CALL GCHAR(YC+1,XC+1,GC)
790 IF GC<>31 THEN 850
800 CR=CR+1
810 IF CR=160 THEN 940
820 CALL CHAR(CR,"")
830 CALL HCHAR(YC+1,XC+1,CR)
840 GC=CR
850 W$=A$(GC-32)
860 MID=(YI-YC*8)*2+1
870 IF XI-INT(XI/8)*8<4 THEN
 890
880 MID=MID+1
890 MID$=SEG$(W$,MID,1)
900 W$=SEG$(W$,1,MID-1)&SEG$
(ST$,POS("0123456789ABCDEF",
MID$,1)*4-3+XI-INT(XI/4)*4,1
)&SEG$(W$,MID+1,16)
910 A$(GC-32)=W$
920 CALL CHAR(GC,W$)
930 RETURN
940 CALL SOUND(1000,440,5)
950 FOR DE=1 TO 100
960 CALL KEY(0,K,S)
970 IF S<>0 THEN 1000
980 NEXT DE
990 GOTO 940
1000 END
```

```
100 ! ************************
110 ! *                      *
120 ! *   STERNENHIMMEL      *
130 ! *    Copyright by      *
140 ! *     Udo Wenzel       *
150 ! *                      *
160 ! *   Bearbeitet vom     *
170 ! *      Team des        *
180 ! *  Aktuell Verlages    *
190 ! *      Muenchen        *
200 ! *                      *
210 ! * Benoetigte Geraete   *
220 ! *  TI99/4A  Konsole    *
230 ! *     Ext. Basic       *
240 ! *                      *
270 ! *  Speicherbelegung    *
280 ! *     3869 Bytes       *
290 ! ************************
295 !
300 CALL CLEAR :: CALL SCREE
N(2):: FOR SCHWARZ=1 TO 14 :
: CALL COLOR(SCHWARZ,16,2)::
 NEXT SCHWARZ
310 DISPLAY AT(1,6)ERASE ALL
:"STERNENHIMMEL" :: DISPLAY
AT(4,1):"GEBEN SIE EINFACH D
IE" :: DISPLAY AT(6,1):"ZAHL
EN FUER DAS ANGEFORDERTE"
320 DISPLAY AT(8,1):"EIN."
330 CALL KEY(0,K,S):: IF S=0
 THEN 310
340 CALL CHARSET
350 PI=ATN(1)*4
360 KI=1
370 KI1=1
380 KI2=1
390 KI3=1
400 CALL CLEAR
410 FOR I=1 TO 14
420 CALL COLOR(I,16,2)
430 NEXT I
440 CALL SCREEN(2)
450 RESTORE 1120
460 CALL CHAR(32,"0")
470 INPUT "BREITENGRAD: ":B
480 IF (B<=-90)+(B>=90)THEN
470
490 INPUT "LAENGENGRAD: ":L
500 IF (L<-180)+(L>180)THEN
490
510 INPUT "JAHR: ":J
520 J=INT(ABS(J))
530 INPUT "MONAT: ":M
540 IF (M<1)+(M>12)THEN 530
550 INPUT "TAG: ":T
560 IF (T<1)+(T>31)THEN 550
570 INPUT "ZEIT(H): ":Z2
580 IF (Z2<1)+(Z2>24)THEN 57
0
590 !
600 GOSUB 1270
610 B=(B*2*PI)/360
620 !
630 K=365*J+M*31+T-46
640 IF M>2 THEN 670
650 K=K+INT((J-1)/4)
660 GOTO 32767
670 K=K-INT(.4*M+2.3)+INT(J/
4)
680 !
690 DEF NA(X)=ATN(X/SQR(-X*X
+1))*KI1
700 DEF NB(X)=KI1*(-ATN(X/SQ
R(-X*X+1)))+PI/2
710 DEF NC(X)=X-INT(X)
720 !
730 AR=(Z2/24*360.9856644+NC
((K-693960)/1461)*1440.02509
)+INT((K-693960)/1461)*.0307
572+99.20189733
740 AR=NC((AR+360)/360)*360
750 !
760 !
770 !
780 FOR O=1 TO 99
790 READ W1,W2
800 W2=(W2*2*PI)/360
810 !
820 SW=L+AR+W1
830 SW=NC((SW+360)/360)*360
840 !
850 SW=(SW*2*PI)/360
860 !
870 H=NA(SIN(B*KI2)*SIN(W2*K
I2)+COS(B*KI3)*COS(W2*KI3)*C
OS(SW*KI3))
880 IF (H*360)/(2*PI)<0 THEN
 1090
890 A=NB((SIN(W2*KI2)-SIN(B*
KI2)*SIN(H*KI2))/(COS(B*KI3)
*COS(H*KI3)))
900 IF SIN(SW*KI2)<0 THEN 92
0
910 A=(2*PI)-A
920 !
930 A=(A*360)/(2*PI)
940 H=(H*360)/(2*PI)
950 !
960 R=(90-H)*500/90
970 W=A+90
```

```
980 W=W/360*(2*PI)
990 X=COS((4*PI-W)*KI3)*R/1.
5
1000 Y=SIN((4*PI-W)*KI2)*R
1010 X1=(X+330)/2.578125
1020 Y1=(Y+515)/5.39
1030 X=X1
1040 Y=Y1
1050 CALL PLOT(X,Y)
1060 !
1070 !
1080 !
1090 NEXT O
1100 CALL SOUND(-1,500,0)
1110 CALL KEY(0,K,S):: IF S=
0 THEN 1110 ELSE 340
1120 DATA 153,49,160,55,166,
56,176,57,194,62,194,56,183,
53
1130 DATA 327,89,272,8,279,7
,273,-9.5,282,-8,274,-2.5,27
6,-1.5,277,-1,50,46
1140 DATA 55,40,67.5,28,64,4
5,48,34
1150 DATA 329,43,346,35.5,35
2,31,358,29,15,28,15,15,358,
15,331,64,338,60,346,61,350,
57,357.5,59
1160 DATA 183,15,192,21,207,
12,200,20,202,24,212,26,215,
23.5
1170 DATA 118,-19,121,-22,12
2,-25,114,-26,109,-33.5,108,
-38,103,-43,97,-43,95,-40,97
.5,-37
1180 DATA 247,32.5,259.5,26,
265,23,262,17,251,22,244,28
1190 DATA 132,33,135,41,143,
38.5,143,31,148,20,139,27
1200 DATA 81,38.5,79,37,77.5
,33,75,32.5,77,37
1210 DATA 263,-18,258,-17,25
3,-26.5,248,-29,256,-28
1220 DATA 165,11.5,167,4,170
,-1,175,0,183,2,163,-6,160,-
11
1230 DATA 314,41,314.5,38.5,
310,49.5,305,47.5,301,40,302
,32
1240 DATA 271,37,271,45,282,
46.5,285.5,44,286.5,33
1250 DATA 74,-5,68,3.5,74,13
.5,63.5,11,62.5,9,57.5,-1
1260 DATA 172,-57,176,-59,17
3,-63,168,-60
1270 CALL CLEAR :: FOR I=33
TO 143 :: CALL CHAR(I,""):
NEXT I :: RETURN
1280 SUB PLOT(X,Y)
1290 IF L=0 THEN L=32 :: H$=
"0123456789ABCDEF"
1300 Y=INT(Y):: X=INT(X):: I
F Y=YO AND X=XO THEN 1460 EL
SE YO=Y :: XO=X
1310 Y1=INT(Y/8):: X1=INT(X/
8):: Y2=Y-Y1*8 :: X3=X-X1*8
:: IF X3<4 THEN X2=1 :: X3=3
-X3 ELSE X2=2 :: X3=7-X3
1320 Z=X2+Y2*2
1330 CALL GCHAR(Y1+1,X1+1,T)
:: IF T<=32 THEN L=L+1 :: T=
L
1340 CALL CHARPAT(T,A$):: B$
=SEG$(A$,Z,1)
1350 C=POS(H$,B$,1)-1 :: C1=
2^X3 :: IF C=0 THEN C=C1 ELS
E 1430
1360 C=C+1
1370 IF Z=1 THEN 1400 ELSE I
F Z=16 THEN 1410
1380 A$=SEG$(A$,1,Z-1)&SEG$(
H$,C,1)&SEG$(A$,Z+1,16-Z)
1390 GOTO 1420
1400 A$=SEG$(H$,1,1)&SEG$(A$
,2,15):: GOTO 1420
1410 A$=SEG$(A$,1,15)&SEG$(H
$,C,1)
1420 CALL CHAR(T,A$):: CALL
HCHAR(Y1+1,X1+1,T):: GOTO 14
60
1430 C3=C :: FOR I=3 TO 0 ST
EP -1 :: C4=2^I1 :: IF C3-C4
>=0 THEN C3=C3-C4 :: IF C4=C
1 THEN 1460
1440 NEXT I
1450 C=C+C1 :: GOTO 1360
1460 SUBEND
```

```
100 ! **********************
110 ! *                    *
120 ! *    ** KLANG **     *
130 ! *    Copyright by    *
140 ! *     G. KAUNTZ      *
150 ! *                    *
160 ! *   Bearbeitet vom   *
170 ! *      Team des      *
180 ! *  Aktuell Verlages  *
190 ! *      Muenchen      *
200 ! *                    *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole  *
230 ! *     Ext. Basic     *
240 ! *                    *
270 ! *  Speicherbelegung  *
280 ! *     1937 Bytes     *
290 ! **********************
295 !
300 CALL CLEAR
310 PRINT "LOESE ALPHA-LOCK
UND DRUECKE.....ENTER.
....." :: INPUT KY$
320 CALL CLEAR
330 T=-1000
340 ON WARNING NEXT
350 L=15 :: ADD=10 :: T1=110
 :: T2=220 :: T3=330 :: R4=-
4
360 DISPLAY AT(1,1):"*******
*********************"
370 DISPLAY AT(2,1):"*.T....
.T.....T.....R.....*"
380 DISPLAY AT(3,1):"*.O....
.O.....O.....A.....*"
390 DISPLAY AT(4,1):"*.N....
.N.....N.....U.....*.*......
.....................*"
400 DISPLAY AT(6,1):"*.1....
.2.....3.....4.....*"
410 DISPLAY AT(7,1):"*******
**********************.*......
.....................*.*******
**********************"
420 DISPLAY AT(8,2)SIZE(6):T
1
430 DISPLAY AT(8,8)SIZE(6):T
2
440 DISPLAY AT(8,14)SIZE(6):
T3
450 DISPLAY AT(8,21)SIZE(3):
R4
460 DISPLAY AT(11,10):"LAUTS
TAERKE =";L
470 DISPLAY AT(13,20):"(L= +
1)"
480 DISPLAY AT(14,20):"(1= -
1)"
490 DISPLAY AT(17,1):"DAUER.
......FREQUENZ-" :: DISPLAY
AT(18,13):"ADDITIV..=";ADD
500 DISPLAY AT(19,20):"(e=1)
"
510 DISPLAY AT(20,20):"(z=10
)"
520 DISPLAY AT(21,1):"(D=+AD
D).....(f=50)" :: DISP
LAY AT(22,1)SIZE(8):"(v= *-1
)"
530 DISPLAY AT(19,1)SIZE(8):
T
540 DISPLAY AT(22,20):"(h=10
0)"
550 DISPLAY AT(24,20):"(t=10
00)"
560 DISPLAY AT(11,1)SIZE(6):
"E=ENDE" :: DISPLAY AT(23,19
)SIZE(9):" (d=30000)"
570 CALL VCHAR(10,11,42,15):
: CALL HCHAR(16,3,42,27)
580 CALL SOUND(T,T1,L,T2,L,T
3,L,R4,L)
590 CALL KEY(0,K,S)
600 IF S=0 THEN 580
610 IF K=49 THEN T1=T1+ADD
620 IF K=33 THEN T1=T1-ADD
630 IF T1<110 OR T1>44000 TH
EN T1=110 :: GOTO 420
640 IF K=50 THEN T2=T2+ADD
650 IF K=64 THEN T2=T2-ADD
660 IF T2<110 OR T2>44000 TH
EN T2=110 :: GOTO 420
670 IF K=51 THEN T3=T3+ADD
680 IF K=35 THEN T3=T3-ADD
690 IF T3<110 OR T3>44000 TH
EN T3=110 :: GOTO 420
700 IF K=52 THEN R4=R4-1
710 IF K=36 THEN R4=R4+1
720 IF R4<-8 OR R4>-1 THEN R
4=-4 :: GOTO 420
730 IF K=101 THEN ADD=1 :: G
OTO 420
740 IF K=122 THEN ADD=10 ::
GOTO 420
750 IF K=102 THEN ADD=50 ::
GOTO 420
760 IF K=104 THEN ADD=100 ::
 GOTO 420
770 IF K=116 THEN ADD=1000 :
: GOTO 420
780 IF K=68 THEN T=T+ADD
790 IF T>4250 OR T<-4250 OR
T=0 THEN T=-100 :: GOTO 420
800 IF K=76 THEN L=L+1
810 IF K=108 THEN L=L-1
820 IF L<0 OR L>30 THEN L=8
:: GOTO 420
830 IF K=100 THEN ADD=30000
:: GOTO 420
840 IF K=69 THEN STOP
850 IF K=118 THEN T=T*-1
860 GOTO 420
```

```
100 REM*********************
110 REM*                    *
120 REM* 3,3 DETERMINANTE *
130 REM*  Copyright  by   *
140 REM*  Andreas Braun   *
150 REM*                    *
160 REM* Bearbeitet  vom  *
170 REM*    Team  des      *
180 REM* Aktuell Verlages *
190 REM*     Muenchen      *
200 REM*                    *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole *
230 REM*                    *
270 REM* Speicherbelegung *
280 REM*    2238 Bytes     *
290 REM*********************
295 REM
300 CALL CLEAR
310 GOTO 370
320 REM PRINTER
330 FOR I=1 TO LEN(A$)
340 CALL HCHAR(Z,I+S,ASC(SEG
$(A$,I,1)))
350 NEXT I
360 RETURN
370 Z=1
380 S=2
390 A$="3,3-DETERMINANTEN NA
CH SARRUS"
400 GOSUB 320
410 CALL CHAR(35,"0000FF")
420 CALL HCHAR(2,3,35,29)
430 Z=4
440 A$="EINGABE DER KOEFFIZI
ENTEN :"
450 GOSUB 320
460 CALL CHAR(36,"1818181818
181818")
470 Z=6
480 A$="$A1 B1 C1$"
490 GOSUB 320
500 Z=7
510 A$="$A2 B2 C2$"
520 GOSUB 320
530 Z=8
540 A$="$A3 B3 C3$"
550 GOSUB 320
560 RESTORE
570 FOR D=1 TO 9
580 READ A$
590 A$=A$&"="
600 DATA A1,A2,A3,B1,B2,B3,C
1,C2,C3
610 SP=0
620 Z=7
630 S=15
640 GOSUB 320
650 A$=""
660 CALL SOUND(100,500,3)
670 CALL KEY(0,K,ST)
680 IF ST=0 THEN 670
690 IF K=13 THEN 890
700 IF K<>7 THEN 750
710 SP=0
720 A$=""
730 CALL HCHAR(Z,18,32,15)
740 GOTO 670
750 IF (K=46)+(K=45)THEN 770
760 IF (K>57)+(K<48)THEN 670
770 SP=SP+1
780 CALL HCHAR(Z,18+SP,K)
790 IF K<>46 THEN 820
800 A$=A$&"."
810 GOTO 860
820 IF K<>45 THEN 850
830 A$="-"
840 GOTO 860
850 A$=A$&CHR$(K)
860 FOR DE=1 TO 80
870 NEXT DE
880 GOTO 670
890 IF A$="" THEN 710
900 KO(D)=VAL(A$)
910 CALL HCHAR(Z,15,32,17)
920 FOR DE=1 TO 120
930 NEXT DE
940 NEXT D
950 Z=10
960 S=2
970 A$="$"&STR$(KO(1))&" "&S
TR$(KO(4))&" "&STR$(KO(7))
980 GOSUB 320
990 Z=11
1000 A$="$"&STR$(KO(2))&" "&
STR$(KO(5))&" "&STR$(KO(8))
1010 GOSUB 320
1020 Z=12
1030 A$="$"&STR$(KO(3))&" "&
STR$(KO(6))&" "&STR$(KO(9))
1040 GOSUB 320
1050 FOR SPA=32 TO 8 STEP -1
1060 FOR ZEI=12 TO 10 STEP -
1
1070 CALL GCHAR(ZEI,SPA,GCH)
1080 IF GCH<>32 THEN 1110
1090 NEXT ZEI
1100 NEXT SPA
1110 CALL VCHAR(10,SPA+1,36,
3)
```

```
1120 ERG=KO(1)*KO(5)*KO(9)+K
O(4)*KO(8)*KO(3)+KO(7)*KO(2)
*KO(6)
1130 ERG=ERG-KO(7)*KO(5)*KO(
3)-KO(1)*KO(8)*KO(6)-KO(4)*K
O(2)*KO(9)
1140 Z=15
1150 S=2
1160 A$="DAS ERGEBNIS : "&ST
R$(ERG)
1170 GOSUB 320
1180 GOTO 1230
1190 Z=20
1200 A$="DRUECKEN SIE EINE B
ELIEBIGE"
1210 GOSUB 320
1220 GOTO 1290
1230 FOR SPAL=32 TO 1 STEP -
1
1240 CALL GCHAR(Z,SPAL,GCH)
1250 IF GCH<>32 THEN 1270
1260 NEXT SPAL
1270 CALL HCHAR(Z+1,3,35,SPA
L-2)
1280 GOTO 1190
1290 Z=22
1300 A$="..TASTE.,.UM EINE W
EITERE"
1310 GOSUB 320
1320 Z=24
1330 A$="DETERMINANTE ZU ERR
ECHNEN !"
1340 GOSUB 320
1350 CALL SOUND(500,500,3)
1360 CALL KEY(0,K,ST)
1370 IF ST=0 THEN 1360
1380 FOR ZEILE=24 TO 9 STEP
-1
1390 CALL HCHAR(ZEILE,1,32,3
2)
1400 NEXT ZEILE
1410 SP=0
1420 GOTO 560
```

DETERMINANTE

Mit Hilfe dieses Programmes gelingt es Ihnen, in äußerst kurzer Zeit die Determinante einer 3,3 Matrix zu ermitteln. Nach dem Start des Programmes stellt der Rechner eine 3x3 Matrix dar und Sie geben im Dialog mit dem Computer die Koeffizienten ein. Wenige Sekunden später können Sie die Determinante der Matrix ablesen.

Andreas Braun

QUADR. ERGÄNZUNG

Seid Ihr gerade dabei, Eure Matheaufgaben zu machen? Dann unterbrecht für 'ne halbe Stunde und seht Euch dieses Programm an. Ihr solltet es vielleicht abtippen, bevor Ihr weitermacht. Anschließend gehen die Aufgaben 100 Mal schneller und Ihr habt noch genügend Zeit für die Disco. Dieses Programm löst quadratische Gleichungen in der Form: $3\ m^2 + 12\ m - 15 = 0$. Und das Schöne dabei ist: Der Computer stellt jeden einzelnen Rechenschritt bis zur Lösung auf dem Bildschirm dar. Alle drei Lösungsmöglichkeiten werden berücksichtigt. Die Ausrechnung erfolgt in Sekundenschnelle.
Von dem sprichwörtlich „langsamen TI99/4A" ist in diesem Programm nichts festzustellen. Und nun viel Spaß beim Aufgaben machen.

M. Steffens

FRANZEZAHL

Lernen Sie gerade Französisch und haben Sie Schwierigkeiten mit den französischen Zahlen?
Der Computer schafft Abhilfe. In den Data-Zeilen dieses Programmes sind alle französischen Zahlen bis zu einer Million enthalten. Nachdem Sie das Programm gestartet haben, werden Sie gefragt, bis zu welcher Obergrenze Sie üben möchten. Z.B. bis 99, die Zahl ist zweistellig, also tippen Sie eine –2– ein. Danach müssen Sie dem Computer nur noch sagen, wieviele Zahlen er Ihnen zum Üben vorgeben soll und schon kann's losgehen. Per Zufallsgenerator gibt der Computer auf dem Bildschirm eine Ziffernfolge bis zu der von Ihnen gewünschten Stelle aus. Übersetzen Sie nun diese Zahl ins Französische. Der Computer sagt Ihnen anschließend, ob Sie die Zahl richtig oder falsch geschrieben haben. Sollte sie falsch gewesen sein, so zeigt er Ihnen die richtige Schreibweise.

B. Knedel

DATEIVERWALTUNG

Da dieses Programm in TI-Basic geschrieben ist, und zudem noch die Auswahlmöglichkeiten Kassettenrekorder, Floppy, Drucker bietet, ist es für jeden TI99/4A-Benutzer anwendbar. Sie können je nach Satzlänge bis zu 200 Datensätze mit jeweils 7 Feldern eingeben. Dieses Programm ist für eine Adreßdatei geschrieben. Sie können es aber auch für jede andere Datei verwenden, indem Sie die Zeilen 320–380 Ihren Bedürfnissen anpassen.
Sie werden von einem Menü durch das ganze Programm geführt, so daß wir hier auf die Steuerung des Programmes nicht weiter einzugehen brauchen. Sollten Sie im Menü die Funktion – Daten abspeichern – vermissen, dies geschieht automatisch, wenn Sie das Programm beenden.
Etwas lästig: Wenn Sie neue Daten eingeben, kehrt das Programm nach jedem Datensatz wieder ins Hauptmenü zurück und Sie müssen erneut die Neuaufnahmefunktion anwählen. Da dieses Programm aber für den privaten Gebrauch bestimmt ist, fällt diese negative Eigenschaft nicht so stark ins Gewicht. Die Vorteile überwiegen eindeutig.
Sie können neue Datensätze hinzufügen, ändern, löschen, sortieren, ausdrucken oder Sie können durch Eingabe eines Feldes alle Datensätze abrufen, die die gleichen Merkmale enthalten, wie das von Ihnen gesuchte Feld.
Alles in Allem: Viel Komfort in wenig Bytes! *B. Knedel*

```
100 REM*********************
110 REM*                    *
120 REM* Quadr.Ergaenzung  *
130 REM*  Copyright  by     *
140 REM*   M. Steffens      *
150 REM*                    *
160 REM* Bearbeitet  vom    *
170 REM*    Team  des       *
180 REM* Aktuell Verlages   *
190 REM*     Muenchen       *
200 REM*                    *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole  *
230 REM*                    *
240 REM* Speicherbelegung  *
250 REM*    4114 Bytes      *
260 REM*********************
270 REM
280 REM BEGINN EINGABETEIL
290 CALL CLEAR
300 CALL SCREEN(4)
310 FOR C=1 TO 13
320 CALL COLOR(C,2,4)
330 NEXT C
340 CALL CHAR(128,"E020E080E
0",131,"0F0808C828281818",13
2,"0000000082442810")
350 PRINT "...ICH LOESE QUAD
RATISCHE.......GLEICHUNGEN D
ER FORM": :
360 PRINT TAB(8);"----------
--"
370 PRINT TAB(8);"¦AX";CHR$(
128);"+BX+C=0¦"
380 PRINT TAB(8);"----------
--": :
390 FOR D=1 TO 500
400 NEXT D
410 PRINT "..GIB NUN DURCH K
OMMATA GE-...TRENNT DIE KOEF
FIZIENTEN.........A,B,C EIN
!": :
420 INPUT A,B,C
430 IF A<>0 THEN 490
440 PRINT : : : : : : : : :
: : : : : : : : : : : : : :
:;"KEINE QUADRATISCHE GLEICH
UNG....!!!!!!!!!!!!!!!!!!!"
: : : : : : : : :
450 FOR D=1 TO 800
460 NEXT D
470 GOTO 290
480 REM ENDE EINGABETEIL
490 REM BEGINN 1. ZEILE
500 A$=STR$(A)&"X"&CHR$(128)
510 IF B>=0 THEN 540
520 B$=STR$(B)&"X"
530 GOTO 550
540 B$="+"&STR$(B)&"X"
550 IF C>=0 THEN 580
560 C$=STR$(C)
570 GOTO 590
580 C$="+"&STR$(C)
590 CALL CLEAR
600 CALL CHAR(129,"007C625F5
555331F",130,"0000247EE77E24
00")
610 S$=CHR$(130)&" "&A$&B$&C
$
620 CALL HCHAR(1,1,129,32)
630 CALL HCHAR(24,1,129,32)
640 G=29
650 Y=3
660 X=0
670 GOSUB 2210
680 S$="=0"
690 X=G
700 GOSUB 2210
710 REM ENDE 1. ZEILE
720 REM BEGINN 2. ZEILE
730 P=B/A
740 Q=C/A
750 A$="X"&CHR$(128)
760 IF P>=0 THEN 790
770 B$=STR$(INT(P*100+0.5)/1
00)&"X"
780 GOTO 800
790 B$="+"&STR$(INT(P*100+0.
5)/100)&"X"
800 IF Q>=0 THEN 830
810 C$=STR$(INT(Q*100+0.5)/1
00)
820 GOTO 840
830 C$="+"&STR$(INT(Q*100+0.
5)/100)
840 S$=CHR$(130)&" "&A$&B$&C
$
850 X=0
860 Y=Y+1
870 GOSUB 2210
880 S$="=0"
890 X=G
900 GOSUB 2210
910 REM ENDE 2.ZEILE
920 REM 3.ZEILE
930 P1=P/2
940 P2=P^2/4
950 B1$="+"&STR$(INT(ABS(P1)
*100+0.5)/100)&CHR$(128)
960 B2$="-"&STR$(INT(P2*100+
0.5)/100)
970 S$=CHR$(130)&" "&A$&B$&B
1$&B2$&C$
980 Y=Y+1
990 X=0
1000 GOSUB 2210
1010 S$="=0"
1020 X=G
1030 GOSUB 2210
1040 REM ENDE 3.ZEILE
1050 REM  BEGINN 4.ZEILE
1060 A$="X"
```

```
1070 IF P>=0 THEN 1100
1080 B3$=STR$(INT(P1*100+0.5
)/100)
1090 GOTO 1110
1100 B3$="+"&STR$(INT(P1*100
+0.5)/100)
1110 S=-P2+Q
1120 IF S>=0 THEN 1150
1130 Q$=STR$(INT(S*100+0.5)/
100)
1140 GOTO 1160
1150 Q$="+"&STR$(INT(S*100+0
.5)/100)
1160 REM
1170 S$=CHR$(130)&" "&"("&A$
&B3$&")"&CHR$(128)&Q$
1180 X=0
1190 Y=Y+1
1200 GOSUB 2210
1210 S$="=0"
1220 X=G
1230 GOSUB 2210
1240 IF S>0 THEN 2040
1250 REM ENDE 4.ZEILE
1260 REM 5.ZEILE
1270 W=SQR(ABS(S))
1280 WL=LEN(STR$(W-INT(W)))
1290 IF WL>4 THEN 1710
1300 W1$="+"&STR$(INT(W*100+
0.5)/100)
1310 W2$="-"&STR$(INT(W*100+
0.5)/100)
1320 S$=CHR$(130)&" "&"("&A$
&B3$&W1$&")("&A$&B3$&W2$&")"
1330 X=0
1340 Y=Y+1
1350 GOSUB 2210
1360 S$="=0"
1370 X=G
1380 GOSUB 2210
1390 REM ENDE 5.ZEILE
1400 REM BEGINN 6.ZEILE
1410 S1=P1+W
1420 S2=P1-W
1430 IF S1>=0 THEN 1460
1440 S1$=STR$(INT(S1*100+0.5
)/100)
1450 GOTO 1470
1460 S1$="+"&STR$(INT(S1*100
+0.5)/100)
1470 IF S2>=0 THEN 1500
1480 S2$=STR$(INT(S2*100+0.5
)/100)
1490 GOTO 1510
1500 S2$="+"&STR$(INT(S2*100
+0.5)/100)
1510 S$=CHR$(130)&" "&"("&A$
&S1$&")("&A$&S2$&")"
1520 X=0
1530 Y=Y+1
1540 GOSUB 2210
1550 S$="=0"
1560 X=G
1570 GOSUB 2210
1580 REM ENDE 6.ZEILE
1590 REM BEGINN 7.ZEILE
1600 X1=-S1
1610 X2=-S2
1620 X1$=STR$(X1)
1630 X2$=STR$(X2)
1640 S$=CHR$(130)&" X="&X1$&
" "&CHR$(132)&" X="&X2$
1650 X=0
1660 Y=Y+1
1670 GOSUB 2210
1680 REM  ENDE 7.ZEILE
1690 IF X1=X2 THEN 1920
1700 GOTO 1980
1710 REM 5.ZEILE W-FALL
1720 W1$=CHR$(131)&STR$(ABS(
INT(S*100+0.5)/100))
1730 W2$=CHR$(131)&STR$(ABS(
INT(S*100+0.5)/100))
1740 S$=CHR$(130)&" ("&A$&B3
$&"+"&W1$&")("&A$&B3$&"-"&W2
$&")"
1750 X=0
1760 Y=Y+1
1770 GOSUB 2210
1780 S$="=0"
1790 X=G
1800 GOSUB 2210
1810 REM ENDE 5.ZEILE W-F.
1820 REM BEGINN 6.ZEILE
1830 X1$=STR$(INT(-P1*100+0.
5)/100)&"-"&W1$
1840 X2$=STR$(INT(-P1*100+0.
5)/100)&"+"&W2$
1850 S$=CHR$(130)&" X="&X1$&
" "&CHR$(132)&" X="&X2$
1860 X=0
1870 Y=Y+1
1880 GOSUB 2210
1890 REM ENDE 6.ZEILE W-F.
1900 GOTO 1980
1910 GOTO 1910
1920 REM FALLS 1 LOESUNG
1930 S$="L={"&X1$&"}"
1940 X=2
1950 Y=Y+2
1960 GOSUB 2210
1970 GOTO 2090
1980 REM FALLS 2 LOESUNGEN
1990 S$="L={"&X1$&";"&X2$&"}
"
2000 X=2
2010 Y=Y+2
2020 GOSUB 2210
2030 GOTO 2090
2040 REM FALLS KEINE LOES.
```

```
2050 S$="L={}"
2060 X=2
2070 Y=Y+2
2080 GOSUB 2210
2090 REM
2100 S$=" EINE WEITERE AUFGA
BE? "
2110 X=4
2120 Y=24
2130 GOSUB 2210
2140 CALL KEY(0,K,STATUS)
2150 IF STATUS=0 THEN 2140
2160 IF (K=74)+(K=106)THEN 2
190
2170 CALL CLEAR
2180 END
2190 CALL CLEAR
2200 GOTO 410
2210 FOR Z=1 TO LEN(S$)
2220 CODE=ASC(SEG$(S$,Z,1))
2230 CALL HCHAR(Y,X+Z,CODE)
2240 NEXT Z
2250 RETURN
```

```
100 REM*********************
110 REM*                    *
120 REM*Franz.Zahlensystem*
130 REM*  Copyright  by     *
140 REM*     B. Knedel     *
150 REM*                   *
160 REM* Bearbeitet  vom    *
170 REM*     Team  des     *
180 REM* Aktuell Verlages  *
190 REM*      Muenchen     *
200 REM*                   *
210 REM*Benoetigte Geraete*
220 REM*                   *
230 REM* TI99/4A  Konsole  *
240 REM*                   *
250 REM* Speicherbelegung  *
260 REM*     3020 Bytes    *
270 REM*********************
280 REM
290 CALL CLEAR
300 RANDOMIZE
310 DIM A$(99)
320 PRINT "...FRANZOESISCHE
ZAHLEN": : : : : : : : : : :
"..HOECHSTGRENZE=1 MILLION"
330 FOR I=1 TO 99
340 READ A$(I)
350 NEXT I
360 CALL CLEAR
370 INPUT "WIEVIEL ZAHLEN ?
":AN
380 PRINT : : : : :
390 INPUT "WELCHE OBERGRENZE
 ? ":OG
400 IF OG>1E6 THEN 670
410 CALL CLEAR
420 FOR I=1 TO AN
430 Z=INT(RND*OG)+1
440 PRINT Z: :
450 PRINT "BITTE ZAHL EINGEB
EN.": :
460 INPUT "":LO$
470 GOSUB 680
480 IF R$=LO$ THEN 560
490 F=F+1
500 PRINT "FALSCH": :
510 PRINT R$: : : : : :
520 FOR Y=1 TO 30
530 CALL SOUND(-100,-6,Y)
540 NEXT Y
550 GOTO 610
560 R=R+1
570 PRINT "RICHTIG": : :
580 FOR Y=1 TO 10
590 CALL SOUND(-50,440,Y)
600 NEXT Y
610 CALL HCHAR(23,1,30,32)
620 PRINT
630 NEXT I
640 PRINT "VON";AN;"ZAHLEN H
ABEN SIE"
```

```
650 PRINT R;"RICHTIGE UND":
:
660 PRINT F;"FALSCHE."
670 END
680 Z$=STR$(Z)
690 ON LEN(Z$)GOSUB 710,710,
730,780,860,940
700 RETURN
710 R$=A$(Z)
720 RETURN
730 X=VAL(SEG$(Z$,1,1))
740 R$=A$(X)&" CENT "
750 X=VAL(SEG$(Z$,2,2))
760 R$=R$&A$(X)
770 RETURN
780 X=VAL(SEG$(Z$,1,1))
790 R$=A$(X)&" MILLE "
800 X=VAL(SEG$(Z$,2,1))
810 IF X=0 THEN 830
820 R$=R$&A$(X)&" CENT "
830 X=VAL(SEG$(Z$,3,2))
840 R$=R$&A$(X)
850 RETURN
860 X=VAL(SEG$(Z$,1,2))
870 R$=A$(X)&" MILLE "
880 X=VAL(SEG$(Z$,3,1))
890 IF X=0 THEN 910
900 R$=R$&A$(X)&" CENT "
910 X=VAL(SEG$(Z$,4,2))
920 R$=R$&A$(X)
930 RETURN
940 X=VAL(SEG$(Z$,1,1))
950 R$=A$(X)&" CENT "
960 X=VAL(SEG$(Z$,2,2))
970 R$=R$&A$(X)&" MILLE "
980 X=VAL(SEG$(Z$,4,1))
990 IF X=0 THEN 1010
1000 R$=R$&A$(X)&" CENT "
1010 X=VAL(SEG$(Z$,5,2))
1020 R$=R$&A$(X)
1030 RETURN
1040 DATA UN,DEUX,TROIS,QUAT
RE,CINQ,SIX,SEPT,HUIT,NEUF,D
IX,ONZE,DOUZE,TREIZE,QUATORZ
E,QUINZE,SEIZE
1050 DATA DIX-SEPT,DIX-HUIT,
DIX-NEUF,VINGT,VINGT ET UN,V
INGT-DEUX,VINGT-TROIS,VINGT-
QUATRE,VINGT-CINQ
1060 DATA VINGT-SIX,VINGT-SE
PT,VINGT-HUIT,VINGT-NEUF,TRE
NTE,TRENTE ET UN,TRENTE-DEUX
,TRENTE-TROIS
1070 DATA TRENTE-QUATRE,TREN
TE-CINQ,TRENTE-SIX,TRENTE-SE
PT,TRENTE-HUIT,TRENTE-NEUF,Q
UARANTE,QUARANTE ET UN
1080 DATA QUARANTE-DEUX,QUAR
ANTE-TROIS,QUARANTE-QUATRE,Q
UARANTE-CINQ,QUARANTE-SIX,QU
ARANTE-SEPT,QUARANTE-HUIT
1090 DATA QUARANTE-NEUF,CINQ
UANTE,CINQUANTE ET UN,CINQUA
NTE-DEUX,CINQUANTE-TROIS,CIN
QUANTE-QUATRE
1100 DATA CINQUANTE-CINQ,CIN
QUANTE-SIX,CINQUANTE-SEPT,CI
NQUANTE-HUIT,CINQUANTE-NEUF,
SOIXANTE,SOIXANTE ET UN
1110 DATA SOIXANTE-DEUX,SOIX
ANTE-TROIS,SOIXANTE-QUATRE,S
OIXANTE-CINQ,SOIXANTE-SIX,SO
IXANTE-SEPT
1120 DATA SOIXANTE-HUIT,SOIX
ANTE-NEUF,SOIXANTE-DIX,SOIXA
NTE ET ONZE,SOIXANTE-DOUZE,S
OIXANTE-TREIZE
1130 DATA SOIXANTE-QUATORZE,
SOIXANTE-QUINZE,SOIXANTE-SEI
ZE,SOIXANTE-DIX-SEPT,SOIXANT
E-DIX-HUIT
1140 DATA SOIXANTE-DIX-NEUF,
QUATRE-VINGTS,QUATRE-VINGT E
T UN,QUATRE-VINGT-DEUX,QUATR
E-VINGT-TROIS
1150 DATA QUATRE-VINGT-QUATR
E,QUATRE-VINGT-CINQ,QUATRE-V
INGT-SIX,QUATRE-VINGT-SEPT,Q
UATRE-VINGT-HUIT
1160 DATA QUATRE-VINGT-NEUF,
QUATRE-VINGT-DIX,QUATRE-VING
T-ONZE,QUATRE-VINGT-DOUZE,QU
ATRE-VINGT-TREIZE
1170 DATA QUATRE-VINGT-QUATO
RZE,QUATRE-VINGT-QUINZE,QUAT
RE-VINGT-SEIZE,QUATRE-VINGT-
DIX-SEPT
1180 DATA QUATRE-VINGT-DIX-H
UIT,QUATRE-VINGT-DIX-NEUF
1190 END
```

```
100 REM*********************
110 REM*                    *
120 REM* Dateiverwaltung    *
130 REM*  Copyright  by     *
140 REM*     B. Knedel      *
150 REM*                    *
160 REM* Bearbeitet  vom    *
170 REM*     Team  des      *
180 REM* Aktuell Verlages   *
190 REM*      Muenchen      *
200 REM*                    *
210 REM*Benoetigte Geraete* *
220 REM* TI99/4A  Konsole   *
230 REM* Kass.Rekorder od.* *
240 REM* Diskettenstation   *
250 REM* Drucker moeglich   *
260 REM*                    *
270 REM* Speicherbelegung   *
280 REM*    3921 Bytes      *
290 REM*********************
295 REM
300 DIM A$(200,7)
310 CALL CLEAR
320 X$(1)="NACHNAME"
330 X$(2)="VORNAME"
340 X$(3)="STRASSE"
350 X$(4)="WOHNORT"
360 X$(5)="GEB.DATUM"
370 X$(6)="TELEFON"
380 X$(7)="BEMERKUNG"
390 F$="CS1"
400 PRINT "WELCHES GERAET BE
NUTZEN SIE?": : : : : :
410 PRINT "1=CASSETTE": : :"
2=DISKETTE"
420 PRINT : : : : :"BITTE ZA
HL EINGEBEN"
430 CALL KEY(0,K,S)
440 IF (K<49)+(K>50)<>0 THEN
 430
450 IF K-48=1 THEN 470
460 F$="DSK1.DAT"
470 CALL CLEAR
480 PRINT "WOLLEN SIE EINE E
RST-DATEI": :"EINGEBEN? J OD
ER N"
490 CALL KEY(0,K,S)
500 IF S=0 THEN 490
510 IF (K=78)+(K=110)THEN 54
0
520 IF (K=74)+(K=106)THEN 60
0
530 GOTO 490
540 OPEN #1:F$,INPUT ,INTERN
AL,FIXED 192
550 INPUT #1:AN
560 FOR Z=1 TO AN
570 INPUT #1:A$(Z,1),A$(Z,2)
,A$(Z,3),A$(Z,4),A$(Z,5),A$(
Z,6),A$(Z,7)
580 NEXT Z
590 CLOSE #1
600 CALL CLEAR
610 PRINT TAB(6);"DATEIVERWA
LTUNG"
620 PRINT : :"1= DATEI ERGAE
NZEN-ERSTELLEN"
630 PRINT : :"2= DATENBEHAND
LUNG"
640 PRINT : :"3= DATENSATZ L
OESCHEN"
650 PRINT : :"4= ALPHABETISC
H SORTIEREN"
660 PRINT : :"5= PROGRAMMEND
E"
670 PRINT : : :"BITTE ZAHL E
INGEBEN"
680 CALL KEY(0,K,S)
690 IF (K<49)+(K>53)<>0 THEN
 680
700 ON K-48 GOTO 790,870,166
0,1880,710
710 CALL CLEAR
720 OPEN #1:F$,OUTPUT,INTERN
AL,FIXED 192
730 PRINT #1:AN
740 FOR Z=1 TO AN
750 PRINT #1:A$(Z,1),A$(Z,2)
,,A$(Z,3),A$(Z,4),A$(Z,5),A$
(Z,6),A$(Z,7)
760 NEXT Z
770 CLOSE #1
780 END
790 CALL CLEAR
800 AN=AN+1
810 FOR Z=1 TO 7
820 PRINT :X$(Z): :
830 INPUT A$(AN,Z)
840 NEXT Z
850 A$(AN,7)=SEG$(A$(AN,7),1
,30)
860 GOTO 600
870 CALL CLEAR
880 PRINT "1= DATEN AUFLISTE
N"
890 PRINT : : :"2= DATEN SUC
HEN"
900 PRINT : : :"3= DATEN AEN
DERN"
910 PRINT : : :"4= HAUPTMENU
E"
920 PRINT : : : : :"BITTE ZA
HL EINGEBEN"
930 CALL KEY(0,K,S)
940 IF (K<49)+(K>52)<>0 THEN
 930
950 ON K-48 GOTO 960,1180,14
50,600
960 CALL CLEAR
970 PRINT "BILDSCHIRM ODER D
RUCKER?"
```

```
980 PRINT : : :"B ODER D"
990 CALL KEY(0,K,S)
1000 IF S=0 THEN 990
1010 IF (K=66)+(K=98)THEN 10
40
1020 IF (K<>68)+(K<>100)THEN
 990
1030 GOTO 2030
1040 CALL CLEAR
1050 FOR Z=1 TO AN
1060 PRINT "SATZNUMMER =";Z:
 :
1070 FOR J=1 TO 7
1080 PRINT A$(Z,J): :
1090 NEXT J
1100 PRINT : :"WEITERLISTEN
J ODER N?"
1110 CALL KEY(0,K,S)
1120 IF (K=74)+(K=106)THEN 1
150
1130 IF (K=78)+(K=110)THEN 6
00
1140 IF S=0 THEN 1110
1150 CALL CLEAR
1160 NEXT Z
1170 GOTO 600
1180 CALL CLEAR
1190 PRINT "WAS WIRD GESUCHT
?": : : :
1200 PRINT : : :"NACHNAME,VO
RNAHME,STRASSE,":"WOHNORT,GE
B.DATUM,TEL.NR.":"ODER BEMER
KUNG"
1210 PRINT : : :"BITTE SUCHW
ORT EINGEBEN.": : :
1220 INPUT P$
1230 PRINT : : :"BITTE WARTE
N."
1240 FOR Z=1 TO AN
1250 FOR J=1 TO 7
1260 IF P$=A$(Z,J)THEN 1350
1270 NEXT J
1280 NEXT Z
1290 CALL CLEAR
1300 PRINT "NICHTS GEFUNDEN!
!!": : : : : : :
1310 CALL SOUND(1000,440,0)
1320 FOR I=1 TO 2000
1330 NEXT I
1340 GOTO 600
1350 CALL CLEAR
1360 PRINT "SATZNR.=";Z: :
1370 FOR D=1 TO 7
1380 PRINT A$(Z,D)
1390 NEXT D
1400 PRINT : : :"WEITERSUCHE
N J ODER N?"
1410 CALL KEY(0,K,S)
1420 IF S=0 THEN 1410
1430 IF (K=78)+(K=110)THEN 6
00
1440 GOTO 1270

1450 CALL CLEAR
1460 PRINT "WELCHER DATENSAT
Z SOLL": : :
1470 PRINT "GEAENDERT WERDEN
?": : :
1480 INPUT O
1490 IF O>AN THEN 1450
1500 CALL CLEAR
1510 FOR J=1 TO 7
1520 PRINT A$(O,J)
1530 NEXT J
1540 PRINT : :"WAS SOLL GEAE
NDERT WERDEN?": :
1550 FOR J=1 TO 7
1560 PRINT J;"= ";X$(J)
1570 NEXT J
1580 PRINT : :"BITTE KENNZAH
L EINGEBEN."
1590 CALL KEY(0,K,S)
1600 IF (K<49)+(K>55)<>0 THE
N 1590
1610 CALL CLEAR
1620 PRINT X$(K-48)
1630 INPUT P$
1640 A$(O,K-48)=P$
1650 GOTO 600
1660 CALL CLEAR
1670 PRINT "WELCHE DATENNUMM
ER SOLL": : :"GELOESCHT WERD
EN?": : :
1680 INPUT P
1690 CALL CLEAR
1700 FOR J=1 TO 7
1710 PRINT A$(P,J)
1720 NEXT J
1730 PRINT : : : : :"LOESCHE
N J ODER N?"
1740 CALL KEY(0,K,S)
1750 IF S=0 THEN 1740
1760 IF (K=74)+(K=106)THEN 1
780
1770 GOTO 600
1780 FOR O=P TO AN-1
1790 FOR J=1 TO 7
1800 A$(O,J)=A$(O+1,J)
1810 NEXT J
1820 NEXT O
1830 FOR J=1 TO 7
1840 A$(AN,J)=""
1850 NEXT J
1860 AN=AN-1
1870 GOTO 600
1880 CALL CLEAR
1890 PRINT "DATEI WIRD SORTI
ERT."
1900 FOR Z=1 TO AN-1
1910 FOR J=1 TO AN-Z
1920 IF A$(J,1)>A$(J+1,1)THE
N 1950
```

```
1930 IF A$(J,1)<>A$(J+1,1)TH
EN 2000
1940 IF A$(J,2)<A$(J+1,2)THE
N 2000
1950 FOR X=1 TO 7
1960 H$=A$(J,X)
1970 A$(J,X)=A$(J+1,X)
1980 A$(J+1,X)=H$
1990 NEXT X
2000 NEXT J
2010 NEXT Z
2020 GOTO 600
2030 OPEN #2:"RS232.LF"
2040 FOR Z=1 TO AN STEP 2
2050 PRINT #2:"SATZNR.=";Z;T
AB(49);"SATZNR.=";Z+1
2060 PRINT #2:CHR$(10)
2070 FOR J=1 TO 7
2080 PRINT #2:X$(J);TAB(14);
A$(Z,J);TAB(50);A$(Z+1,J)
2090 NEXT J
2100 PRINT #2:CHR$(10);CHR$(
10)
2110 NEXT Z
2120 CLOSE #2
2130 GOTO 600
```

VIDEOTHEK

Dies ist ein Archivierungsprogramm, speziell für Sammler von Videofilmen konzipiert. Mit der maximalen Konfiguration lassen sich 120 Filme nach 5 verschiedenen Kategorien archivieren und bearbeiten. Wer nicht im Besitz einer Speichererweiterung ist, muß die Dimensionierung entsprechend ändern. Als externes Speichermedium ist die Diskettenstation vorgesehen, da die extrem langen Verarbeitungszeiten des Kassettenrekorders für ein Dateiprogramm nicht geeignet sind. Dank einer übersichtlichen Benutzerführung werden Sie sich schnell an dieses Programm gewöhnt haben. Sie können jederzeit erkennen, in welchem Teil des Menüs Sie sich befinden und welcher Arbeitsvorgang gerade abläuft.

PROGRAMMBESCHREIBUNG:

Nach dem Start des Programmes mit "RUN" generiert der Computer ein Titelbild, die "ENTER"-Taste führt Sie nun zum Hauptmenü. Von hier kommen Sie durch Eingabe der jeweiligen Kennziffer (ohne ENTER) in jeden Teil des Programmes.

1. DATEN VON DISKETTEN LADEN

Ihre bis zu diesem Zeitpunkt archivierten Filme werden in den Computer geladen. In einer Übersicht können Sie zur gleichen Zeit ablesen, wieviele Filme Sie schon archiviert wurden und welche Dateinummer der Rechner gerade lädt. Nach erfolgreich durchgeführtem Ladevorgang kommen Sie in das Hauptmenü zurück.

2. DATEN SUCHEN

Sie können nun in der Auswahlliste eine bestimme Kategorie wählen, die der Rechner durchsuchen soll. Nach Wahl der Kategorie kontrolliert der Rechner, ob die Datei nach dieser Kategorie sortiert ist. Ist dies nicht der Fall, so wird sie umsortiert und damit der anschließende Suchvorgang beschleunigt. Bei einem vollen Archiv (120 Filme) hat der Rechner nach maximal 7 Zugriffen den Datensatz mit dem von Ihnen gewünschten Stichwort gefunden.
Die ENTER-Taste führt Sie nun in die Wahlliste 2, in der Ihnen weitere Verarbeitungsmöglichkeiten angeboten werden.

3. ARCHIV

Führt Sie ebenfalls in Wahlliste 2.

4. NEUE DATEN EINGEBEN

In diesem Teil können weitere Filme eingegeben werden. Titel, Kassettennummer, Filmstar, Filmart- und Länge werden nacheinander eingegeben. Die ENTER-Taste bei "TITEL" führt ins Hauptmenü zurück. Drücken Sie die ENTER-Taste in einem der anderen 4 Felder, bekommen Sie eine Leerzeile.

5. DATEN SPEICHERN

In diesem Teil werden Ihre Daten auf Diskette zurückgespeichert. Dateileichen werden zuvor entfernt.

6. DATEN AUSDRUCKEN

Sie können die gesamte Datei oder auch einzelne Datensätze ausdrucken lassen. Sollten Sie nicht über einen Thermodrucker verfügen, müssen die Druckersteuerungsbefehle entsprechend umgeschrieben werden.

7. DIAGRAMM

Der Rechner erstellt Ihnen eine dreidimensionale Statistik für Ihre Filme, aufgeteilt in die einzelnen Filmarten. Die Enter-Taste führt ins Hauptmenü zurück.

8. ENDE

Nur durch Wahl der Kennziffer 8 im Hauptmenü kann das Programm beendet werden. Zuvor fragt der Rechner noch, ob auch alle Daten gesichert sind.

MENÜ 2

In diesem Menü stehen Ihnen 4 verschiedene Verarbeitungsmöglichkeiten zur Verfügung. Punkt 5 bringt das Hauptmenü zurück.

1. DATEN VERÄNDERN

Der Computer fragt nach der Dateinummer des zu ändernden Datensatzes, anschließend wird Ihnen der Satz ausgegeben. Nach dem Korrekturvorgang kommen Sie automatisch in Wahlliste 2 zurück.

2. DATEN AUSDRUCKEN

Hat die gleiche Funktion wie Punkt 6 im Hauptmenü.

3. BLÄTTERN

Sie können in Ihrem Archiv vor- und rückwärts blättern. Geben Sie nur die Dateinummer an, mit der Sie beginnen möchten.

4. SORTIEREN

Durch Eingabe der Kennziffer können Sie Ihr Archiv nach jeder der 5 Kategorien sortieren lassen. Nach erfolgtem Sortiervorgang verzweigt das Programm wieder ins zweite Menü.

Hinweise zum Abtippen des Programmes:

Aus Optimierungsgründen sind manche Programmzeilen länger als 140 Bytes, dann schließen Sie die Zeile mit 'ENTER' (offene Anführungszeichen müssen zuvor geschlossen werden) und holen Sie sich mit 'FCTN REDO' zurück. Jetzt kann der Rest der Zeile angefügt werden.

Hartmut König

```
100 ! ************************
120 ! *     VIDEOTHEK        *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole   *
230 ! *      Ext. Basic     *
240 ! *  Diskettenstation   *
260 ! *        Drucker      *
290 ! ************************
295 !
300 CALL TITEL
310 CALL CHAR(95,"0000FFFFFF
FF"):: HL$=RPT$("_",28):: CA
LL SCREEN(8):: FOR I=1 TO 12
 :: CALL COLOR(I,2,16):: NEX
T I
320 ON WARNING NEXT :: ON BR
EAK NEXT
330 FILM$(1)="TITEL" :: FILM
$(2)="KASSETTE" :: FILM$(3)=
"FILMSTAR" :: FILM$(4)="FILM
ART" :: FILM$(5)="LAENGE"
340 DIM V$(5,120),SU(12),P$(
12),SR(12)
350 CALL VCHAR(1,1,32,768)::
 DISPLAY AT(1,1):HL$ :: DISP
LAY AT(2,8):"VIDEOARCHIV" ::
 DISPLAY AT(3,1):HL$ :: DISP
LAY AT(6,9):"WAHLLISTE":"...
.....__________"
360 DISPLAY AT(10,1):"1-> DA
TEN VON DISKETTE LADEN": :"2
-> DATEN SUCHEN": :"3-> ARCH
IV"
370 DISPLAY AT(16,1):"4-> NE
UE DATEN EINGEBEN": :"5-> DA
TEN SPEICHERN": :"6-> DATEN
AUSDRUCKEN": :"7-> DIAGRAMM"
: :"8-> E N D E"
380 CALL KEY(0,K,S):: IF S=0
 THEN 380 ELSE IF K<49 OR K>
56 THEN 380 ELSE CALL VCHAR(
1,1,32,768):: ON K-48 GOTO 3
85,420,530,620,585,570,970,8
80
385 ON ERROR 350
390 OPEN #1:"DSK1.VIDARCHIV"
,INTERNAL,INPUT ,FIXED 90,RE
LATIVE 120 :: DISPLAY AT(10,
5):"BITTE WARTEN...!!"
400 INPUT #1,REC 0:A,C :: DI
SPLAY AT(15,5):"FILME: ";STR
$(A):"....FREI :";120-A :: D
ISPLAY AT(20,5):"DATEI IST N
ACH ";FILM$(C):"....SORTIERT
"
410 FOR D=1 TO A :: INPUT #1
,REC D:V$(1,D),V$(2,D),V$(3,
D),V$(4,D),V$(5,D):: DISPLAY
 AT(15,15):"A";STR$(D);"Ü" :
: NEXT D :: CLOSE #1 :: GOTO
 350
420 DISPLAY AT(1,1):HL$:"...
....WAHLLISTE-SUCHEN" :: DIS
PLAY AT(3,1):HL$ :: FOR I=1
TO 5 :: DISPLAY AT(I+8,3):I;
"-> ";FILM$(I):: NEXT I
430 DISPLAY AT(7,3):"SUCHEN
NACH:"
440 CALL KEY(0,K,S):: IF S=0
 THEN 440 ELSE IF K<49 OR K>
53 THEN 440 ELSE T1=K-48
450 IF T1<>C THEN C=T1 :: DI
SPLAY AT(20,3):"AUGENBLICK B
ITTE..." :: CALL SHELLSORT(V
$(,),A,C):: DISPLAY AT(20,3)
:""
460 CALL HCHAR(4,1,32,664)::
 DISPLAY AT(1,1):HL$ :: DISP
LAY AT(2,3):C;FILM$(C)&" SUC
HEN":HL$: : :"BITTE ";FILM$(
C);" EINGEBEN" :: ACCEPT AT(
8,1):M$
470 K=0 :: LM=LEN(M$):: I=A
480 J=INT((I+K)/2):: IF C=2
THEN 500
490 IF M$>SEG$(V$(C,J),1,LM)
THEN K=J :: GOTO 510 :: ELSE
 I=J :: GOTO 510
500 IF V$(2,J)="" THEN 350 E
LSE IF VAL(M$)>VAL(V$(2,J))T
HEN K=J ELSE I=J
510 IF K+1<I THEN 480
520 CALL HCHAR(4,1,32,320)::
 GOSUB 680 :: CALL VCHAR(1,1
,32,768)
530 DISPLAY AT(1,1):HL$:"...
..WAHLLISTE - 2":HL$
540 DISPLAY AT(6,3):"1 -> DA
TEN VERAENDERN": :"..2 -> DA
TEN AUSDRUCKEN": :"..3 -> BL
AETTERN": :"..4 -> SORTIEREN
": :"..5 -> WEITER"
550 CALL KEY(0,K,S):: IF S=0
 THEN 550 ELSE IF K<49 OR K>
53 THEN 550 ELSE ON K-48 GOT
O 710,570,740,1050,560
560 CALL VCHAR(1,1,32,768)::
 GOTO 350
570 DISPLAY AT(1,1):HL$:"...
..DATEN AUSDRUCKEN":HL$ :: C
```

```
ALL HCHAR(4,1,32,640):: DISP
LAY AT(23,3):"IST EIN DRUCKE
R":"..ANGESCHLOSSEN?<J/N>"
580 CALL KEY(0,K,S):: IF S=0
 THEN 580 ELSE IF K<>74 OR K
<>106 THEN 530 ELSE 800
585 ON ERROR 350
590 OPEN #1:"DSK1.VIDARCHIV"
,INTERNAL,OUTPUT,FIXED 90,RE
LATIVE 120
595 DISPLAY AT(1,1):HL$:"...
....DATEN SPEICHERN":HL$ ::
DISPLAY AT(12,2):"FILMNUMMER
"
600 GOSUB 940 :: PRINT #1,RE
C 0:A,C :: FOR D=1 TO A :: D
ISPLAY AT(12,13):"A";STR$(D)
;"Ü"
610 PRINT #1,REC D:V$(1,D),V
$(2,D),V$(3,D),V$(4,D),V$(5,
D):: NEXT D :: CLOSE #1 :: C
ALL CLEAR :: GOTO 350
620 DISPLAY AT(1,1):HL$ :: D
ISPLAY AT(2,8):"NEUAUFNAHME"
 :: DISPLAY AT(3,1):HL$ :: D
ISPLAY AT(8,1):"BEENDEN DER
EINGABE DURCH"
630 DISPLAY AT(9,1):">ENTER<
 IN SEKTION..'TITEL'":">ENTE
R< IN ZEILE 2-5 ERGIBT":"LEE
RZEILE" :: FOR DEL=1 TO 1200
 :: NEXT DEL :: CALL HCHAR(5
,1,32,640)
640 A=A+1 :: FOR I=1 TO 5 ::
 PX=PX+2 :: DISPLAY AT(I+5+P
X,1):FILM$(I)
650 ACCEPT AT(I+6+PX,1)SIZE(
28):V$(I,A):: IF V$(1,A)=""
THEN A=A-1 :: C=0 :: GOTO 67
0
660 NEXT I :: PX=0 :: GOTO 6
40
670 CALL CLEAR :: GOTO 350
680 DISPLAY AT(10,1):USING "
DATEINUMMER ##":I;:: FOR J=1
 TO 5 :: DISPLAY AT(J+11,1):
V$(J,I):: NEXT J
690 CALL KEY(0,K,S):: IF S=0
 THEN 690
700 RETURN
710 DISPLAY AT(1,1):HL$:"...
MODUS 13 - AENDERUNG":HL$ ::
 DISPLAY AT(23,1):"WELCHE DA
TEINUMMER ?" :: ACCEPT AT(23
,22)VALIDATE(DIGIT)SIZE(-3):
I :: NT=4
720 FOR J=1 TO 5 :: NT=NT+2
:: DISPLAY AT(NT,1):V$(J,I):
: NEXT J :: NT=4 :: FOR J=1
TO 5 :: NT=NT+2 :: ACCEPT AT
(NT,1)SIZE(-28)BEEP:V$(J,I):
: NEXT J
730 CALL VCHAR(1,1,32,768)::
 GOTO 530
740 CALL CLEAR :: DISPLAY AT
(1,1):HL$:"3...-....KATALOG"
:HL$
750 DISPLAY AT(10,2):"AB WEL
CHER DATEINUMMER":"WOLLEN SI
E BLAETTERN ?" :: ACCEPT AT(
14,13)VALIDATE(DIGIT)SIZE(3)
BEEP:I
760 CALL HCHAR(4,1,32,640)::
 DISPLAY AT(8,9):"BLAETTERN"
:".........__________" :: DISP
LAY AT(12,3):"1-> VORWAERTS"
: :"..2-> RUECKWAERTS": :"..
3-> BLAETTERN ENDE"
770 CALL KEY(0,K,S):: IF S=0
 THEN 770 ELSE IF K=51 THEN
790 ELSE IF K=50 THEN IF I>=
3 THEN I=I-2 ELSE I=3 ELSE I
F K<>49 THEN 770
780 N=8 :: CALL HCHAR(8,1,32
,182):: DISPLAY AT(7,7):"DAT
EINUMMER ";I :: FOR J=1 TO 5
 :: N=N+2 :: DISPLAY AT(N,1)
:V$(J,I):: NEXT J :: I=I+1 :
: GOTO 770
790 CALL HCHAR(4,1,32,640)::
 GOTO 530
800 DISPLAY AT(1,1):HL$:"...
..DATEN AUSDRUCKEN":HL$ :: C
ALL HCHAR(4,1,32,640)
810 DISPLAY AT(8,1):"1-> EIN
E BESTIMMTE DATEI-":"....NUM
MER AUSDRUCKEN": :"2-> ALLE
DATEINUMMERN":"....AUSDRUCKE
N" :: DISPLAY AT(20,1)BEEP:"
BITTE WAEHLEN SIE !"
820 CALL KEY(0,K,S):: IF S=0
 THEN 820 ELSE IF K=49 THEN
830 ELSE IF K=50 THEN 840 EL
SE 820
830 DISPLAY AT(15,7)BEEP:"DA
TEINUMMER ?" :: ACCEPT AT(17
,13)VALIDATE(DIGIT)SIZE(3):I
 :: GOSUB 845 :: GOTO 530
```

```
840 FOR I=1 TO A :: GOSUB 84
5 :: NEXT I :: GOTO 530
845 ON ERROR 870
850 DISPLAY AT(23,1):"STARTZ
EILE,ENDZEILE(1-5)=?.." :: A
CCEPT AT(23,26):SZ :: ACCEPT
 AT(23,28):EZ :: OPEN #2:"TP
",OUTPUT
860 PRINT #2:STR$(I):HL$ ::
FOR X=SZ TO EZ :: PRINT #2:V
$(X,I):: NEXT X :: PRINT #2:
HL$ :: CLOSE #2
870 CALL CLEAR :: RETURN
880 DISPLAY AT(6,7):" A C H
T U N G":".......___________
__"
890 DISPLAY AT(10,2):"WENN S
IE DAS PROGRAMM BE-":"ENDEN
WERDEN ALLE DATEN":"GELOESCH
T !!!"
900 DISPLAY AT(18,9):"WAHLLI
STE":".........__________" ::
DISPLAY AT(21,3):"1-> E N D
E" :: DISPLAY AT(23,3)BEEP:"
2-> W E I T E R"
910 CALL KEY(0,K,S):: IF S=0
 THEN 910 ELSE IF K=49 THEN
930 ELSE IF K=50 THEN 920 EL
SE 910
920 CALL CLEAR :: GOTO 350
930 CALL CLEAR :: END
940 L=A :: A=1 :: FOR J=1 TO
 L :: IF V$(1,J)="" THEN 960
950 V$(1,A)=V$(1,J):: V$(2,A
)=V$(2,J):: V$(3,A)=V$(3,J):
: V$(4,A)=V$(4,J):: V$(5,A)=
V$(5,J):: A=A+1
960 NEXT J :: A=A-1 :: RETUR
N
970 DISPLAY AT(10,5)ERASE AL
L:"BITTE WARTEN...!!": :"...
.DIAGRAMM WIRD ERSTELLT"
980 RESTORE
990 DATA ACTION,ABENTEUER,"S
IENCE FICTION",PORNO,KRIMI,K
RIEGSFILM,SPIELFILM,HORROR,K
ARATE,COMIC,KATASTROPHE,ROCK
ER
1000 FOR J=1 TO 12 :: READ P
$(J):: FOR I=1 TO A :: IF V$
(4,I)=P$(J)THEN BE=BE+1 :: S
U(J)=BE
1010 NEXT I
1020 BE=0 :: NEXT J
1030 DATA AC,AB,SF,PO,TR,KR,
SP,HO,KT,CO,KS,RO
1040 FOR J=1 TO 12 :: READ P
$(J):: NEXT J :: ANZ=12 :: C
ALL DIAGRAMM((ANZ),SU(),P$()
,SR()):: GOTO 310
1050 CALL CLEAR :: DISPLAY A
T(1,1):HL$:".........SORTIER
EN":HL$ :: DISPLAY AT(6,3):"
SORTIEREN NACH:"
1060 FOR I=1 TO 5 :: I1$=STR
$(I):: DISPLAY AT(I+7,3):I1$
&"->.."&FILM$(I):: NEXT I ::
 DISPLAY AT(16,3):"IHRE WAHL
?"
1070 ACCEPT AT(16,13)VALIDAT
E("12345")SIZE(1):C :: DISPL
AY AT(22,1):"DATEI WIRD NACH
 ";FILM$(C);" SORTIERT" :: I
F C=2 THEN GOSUB 940
1080 CALL SHELLSORT(V$(,),A,
C):: CALL CLEAR :: GOTO 530
1090 SUB TITEL
1100 CALL CLEAR :: CALL CHAR
SET :: CALL SCREEN(2):: FOR
I=1 TO 14 :: CALL COLOR(I,10
,1):: NEXT I
1110 A$(1)="................
................................
................................
..............+..+.+.++b.++
+.a+b......."
1120 A$(2)="...+..+.+.+.+.+.
..+.+..........+..+.+.+.+.++
+.+.+.+++......egfd.+.+.+.+.
..+.+..........ed..+.++§.++
+.>+§......."
1130 A$(3)="................
................................
................................
..............a+b.++b.a+b.+
.+.+.+..+..."
1140 A$(4)="...+.+.+.+.+.+.+.+
.+.+.+..+.....+++.++§.+...+
++.+.+..+.....+.+.+<..+.+.+
.+.+.egfd.....+.+.+eg.>+§.+
.+.+..ed...."
1150 A$(5)="................
................................
..............++++++++++++
++++++++++.................
............"
```

```
1160 A$(6)="....a++++++++++
++++++++b.......+.COPYRIGHT.
1984.BY.+.......+..HARTMUT..
KOENIG..+.......>§..........
.......>§..."
1170 CALL CHAR(42,RPT$("0",1
6),43,RPT$("F",16),47,"FF818
181818181FF",60,"F0F0F8FCFCF
EFFFF")
1180 CALL CHAR(62,"FFFF7F7F1
F1F0707",64,"FFFFFEFEF8F8E0E
0",97,"07071F1F7F7FFFFF",98,
"E0E0F8F8FEFEFFFF")
1190 CALL CHAR(100,"FFFEFCF8
F0E0C080",101,"FF7F3F1F0F070
301",102,"0103070F1F3F7FFF",
103,"80C0E0F0F8FCFEFF")
1200 DISPLAY AT(1,1):A$(1);A
$(2);A$(3);A$(4);A$(5);A$(6)
:: CALL HCHAR(24,1,43,64)::
CALL VCHAR(1,31,43,96)
1210 CALL KEY(0,K,S):: IF S=
0 THEN 1210
1220 CALL CLEAR :: CALL CHAR
SET :: SUBEND
1230 !
1240 SUB SHELLSORT(S$(,),N,C
)
1250 M=N-1
1260 M=INT(M*.3+.5)
1270 FOR K=1 TO M :: FOR I=K
+M TO N STEP M
1280 X$(1)=S$(1,I):: X$(2)=S
$(2,I):: X$(3)=S$(3,I):: X$(
4)=S$(4,I):: X$(5)=S$(5,I)
1290 FOR J=I-M TO 1 STEP -M
:: IF C<>2 THEN IF X$(C)>=S$
(C,J)THEN 1340 ELSE 1310
1300 IF C=2 THEN IF VAL(X$(C
))>=VAL(S$(C,J))THEN 1340
1310 Y=J+M :: S$(1,Y)=S$(1,J
):: S$(2,Y)=S$(2,J):: S$(3,Y
)=S$(3,J)
1320 S$(4,Y)=S$(4,J):: S$(5,
Y)=S$(5,J)
1330 NEXT J
1340 Y=J+M :: S$(1,Y)=X$(1):
: S$(2,Y)=X$(2):: S$(3,Y)=X$
(3)
1350 S$(4,Y)=X$(4):: S$(5,Y)
=X$(5)
1360 NEXT I :: NEXT K
1370 IF M>1 THEN 1260
1380 SUBEND
1390 ! ====================
1400 !
1410 SUB DIAGRAMM(AN,A(),P$(
),SR()):: CALL CHARSET :: A$
="0103070F1F3F7FFF" :: CALL
CLEAR
1420 CALL CHAR(96,RPT$("F",1
6),104,RPT$("F",16),105,A$,1
12,A$,121,A$,130,"FFF7F3F1F0
703010")
1430 CALL COLOR(9,4,1,10,5,2
,11,11,5,12,2,8,13,1,8):: FE
HLER=0 :: IF AN<1 OR AN>12 T
HEN SUBEXIT ELSE ZEIGER=0
1440 FOR I=1 TO AN :: SR(I)=
A(I):: NEXT I :: CALL HCHAR(
4,1,32,500):: FOR I=1 TO AN
:: IF A(I)>AMAX THEN AMAX=A(
I)
1450 NEXT I :: AMAXR=20/AMAX
 :: FOR I=1 TO AN :: A(I)=IN
T(A(I)*AMAXR):: NEXT I :: FO
R I=1 TO AN :: SP=I*2+4 :: I
F A(I)<1 THEN 1480
1460 HO=25-A(I):: SPI=HO-1 :
: CALL VCHAR(HO,SP,96,A(I)):
: CALL HCHAR(SPI,SP,121):: C
ALL VCHAR(HO,SP+1,104,A(I)-1
)
1470 CALL HCHAR(SPI,SP+1,105
):: CALL HCHAR(24,SP+1,112):
: GOTO 1490
1480 CALL HCHAR(24,SP,121)::
 CALL HCHAR(24,SP+1,130)
1490 NEXT I :: CALL COLOR(3,
2,4,4,2,4)
1500 FOR I=1 TO AN :: DISPLA
Y AT(1,I*2+3):SEG$(P$(I),1,1
):: DISPLAY AT(2,I*2+3):SEG$
(P$(I),2,1):: NEXT I
1510 FOR I=1 TO AN :: SP=I*2
+2 :: AD$=STR$(SR(I)):: IF L
EN(AD$)=1 THEN 1520 ELSE 153
0
1520 DISPLAY AT(24,SP)SIZE(1
):AD$ :: GOTO 1540
1530 DISPLAY AT(23,SP)SIZE(1
):SEG$(AD$,1,1):: DISPLAY AT
(24,SP)SIZE(1):SEG$(AD$,2,1)
1540 NEXT I
1550 CALL KEY(0,K,S):: IF S=
0 THEN 1550 ELSE CALL HCHAR(
1,1,32,768):: CALL CHARSET
1560 SUBEND
```

```
100 REM**********************
110 REM*                    *
120 REM*TABELLENERSTELLUNG*
130 REM*  Copyright by      *
140 REM*  Andreas Braun     *
150 REM*                    *
160 REM* Bearbeitet vom     *
170 REM*    Team des        *
180 REM* Aktuell Verlages    *
190 REM*     Muenchen       *
200 REM*                    *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole   *
230 REM*                    *
270 REM* Speicherbelegung   *
280 REM*    2712 Bytes      *
290 REM**********************
295 REM
300 DIM B(300),WERT(300),PL(
300),NAME$(300),AZ(300)
310 CALL CHAR(63,"00C0F0FCFF
FCF0C0",35,"FFFFFFFFFFFFFFFF
")
320 CALL CLEAR
330 PRINT "WONACH WIRD GEWER
TET ?": :"1- NACH GERINGEM E
RGEBNIS": :"2- NACH HOHEM ER
GEBNIS ": :
340 INPUT WE
350 IF (WE=1)+(WE=2)THEN 360
 ELSE 330
360 CALL CLEAR
370 A=0
380 PRINT "##################
############ EINGABE DER STA
RTER UND..##.............
.........##......DEREN ERGE
BNISSE....##################
############": :
390 A=A+1
400 IF (INT(A/6)=A/6)+(A=1)T
HEN 410 ELSE 420
410 PRINT "##################
############STAND! ENDE! VOR
! ZURUECK!##RUNDE! (=NAECHST
E RUNDE ) ##################
############"
420 PRINT "STARTNUMMER ";A;
430 IF NAME$(A)="" THEN 450
440 PRINT TAB(1);"EINGABE:'E
NTER'=A"&NAME$(A)&"Ü":TAB(16
);
450 B(A)=A
460 NAME1$=NAME$(A)
470 INPUT NAME$(A)
480 IF NAME$(A)<>"" THEN 500
490 NAME$(A)=NAME1$
500 IF NAME$(A)<>"RUNDE!" TH
EN 570
510 ASF=A-1
520 FOR QWE=1 TO ASF
530 AZ(QWE)=WERT(QWE)
540 NEXT QWE
550 A=0
560 GOTO 380
570 IF NAME$(A)="ENDE!" THEN
 770
580 IF NAME$(A)="STAND!" THE
N 770
590 IF NAME$(A)<>"ZURUECK!"
THEN 640
600 INPUT "ZU WELCHER STARTN
UMMER?":ZUR
610 MEMORY=A
620 A=ZUR
630 GOTO 400
640 IF NAME$(A)<>"VOR!" THEN
 720
650 NAME$(A)=NAME1$
660 INPUT "ZU WELCHER STARTN
UMMER.:....EINGABE ZUR VORLE
TZTEN'0'?":VOR
670 IF VOR<>0 THEN 700
680 A=MEMORY
690 GOTO 400
700 A=VOR
710 GOTO 400
720 PRINT "DER WERT VON";A;
730 INPUT WERT(A)
740 WERT(A)=WERT(A)+AZ(A)
750 PRINT
760 GOTO 390
770 STA=A-1
780 ON WE GOTO 790,870
790 FOR I=1 TO STA
800 PL(I)=1
810 FOR N=1 TO STA
820 IF WERT(I)<=WERT(N)THEN
840
830 PL(I)=PL(I)+1
840 NEXT N
850 NEXT I
860 GOTO 940
870 FOR I=1 TO STA
880 PL(I)=1
890 FOR N=1 TO STA
900 IF WERT(I)>=WERT(N)THEN
920
910 PL(I)=PL(I)+1
920 NEXT N
```

```
930 NEXT I
940 CALL CLEAR
950 IF NAME$(A)<>"STAND!" TH
EN 980
960 PRINT "##################
############ STAND NACH";A-1
;"STARTERN";TAB(28);"#":"###
##########################":
:
970 GOTO 990
980 PRINT "##################
############..E N D E R G E
B N I S...##################
############": :
990 FOR PLA=1 TO STA
1000 FOR I=1 TO STA
1010 IF PL(I)<>PLA THEN 1110
1020 H$=STR$(PL(I))
1030 I$=STR$(B(I))
1040 PRINT H$;TAB(4);I$;TAB(
7);NAME$(I);TAB(22);WERT(I)
1050 CALL KEY(0,KEY,STATUS)
1060 IF STATUS=0 THEN 1110
1070 FOR DE=1 TO 500
1080 NEXT DE
1090 CALL KEY(0,KEY,STATUS)
1100 IF STATUS=0 THEN 1090
1110 NEXT I
1120 NEXT PLA
1130 CALL KEY(0,KEY,STATUS)
1140 IF STATUS=0 THEN 1130
1150 PRINT "_______________
_____________DRUECKE: -'1' ZU
M ERGEBNIS.....-'2' ZU
M FORTFAHREN.....-'3' ZU
R NEUEINGABE.....-'4' ZU
M NEUANFANG"
1160 PRINT ".....-'5' ZU
M ABSCHLUSS.....-'6' ZU
R NAECHSTEN.....
..RUNDE"
1170 CALL KEY(0,KEY,STATUS)
1180 IF STATUS=0 THEN 1170
1190 IF (KEY<49)+(KEY>54)THE
N 1170
1200 PRINT : :
1210 IF (KEY=51)+(KEY=52)THE
N 1220 ELSE 1270
1220 FOR JI=1 TO 300
1230 NAME$(JI)=""
1240 WERT(JI)=0
1250 AZ(JI)=0
1260 NEXT JI
1270 ON KEY-48 GOTO 940,420,
360,320,1280,510
1280 INPUT "BEENDIGUNG DURCH
..'E':A$
1290 IF A$<>"E" THEN 1150
1300 END
```

TABELLENERSTELLUNG

Dieses Programm bietet Ihnen die Möglichkeit, Tabellen und Ranglisten jeder Art zu erstellen. Sie können bis zu 300 Teilnehmer mit diesem Programm überwachen und eingliedern.

PROGRAMMSTEUERUNG:

Nach dem Programmstart wird als erstes die Art der Tabelle angefordert. Geben Sie '1' für aufsteigend und '2' für absteigend ein.

Jetzt können die Teilnehmer und deren Zeiten oder Ergebnisse eingegeben werden. Der Computer sortiert die Eingaben in der von Ihnen gewünschten Reihenfolge.

Die Eingabe kann jederzeit mit einem der folgenden Befehle unterbrochen oder beendet werden:

STAND! =	gibt den augenblicklichen Tabellenstand als Zwischenergebnis aus.
ZURUECK!=	Sie können mit der Eingabe bei einer niedrigen Tabellennummer Ihrer Wahl neu beginnen.
VOR! =	mit nachfolgendem 'O' zeigt die vorletzte Startnummer an. Ohne 'O' springt das Programm zu einer höheren Startnummer Ihrer Wahl.
RUNDE! =	Die Ergebnisse der nächsten Runde werden mit denen der letzten Runde addiert.
ENDE! =	Sie können das Programm durch Eingabe von 'E' beenden.

Das Ausrufezeichen hinter dem Befehlswort muß unbedingt folgen, da der Befehl sonst als Teilnehmername verwendet wird. Sollten Sie die Ergebnisse dauerhaft auf Diskette oder Kassette speichern wollen, wird es Ihnen sicherlich nicht schwer fallen, diese kleine Routine noch in das Programm einzufügen.

Andreas Braun

JAHRESZINSEN

Mit diesem Programm können Sie die voraussichtlichen Gesamtzinsen des laufenden Jahres berechnen, am Jahresende die tatsächlichen Zinsen des vergangenen Jahres. Das Programm arbeitet nur mit Eingaben zwischen dem 1.1. und 31.12. einschließlich eines Übertrages aus dem vergangenen Jahr.

100 verschiedene Buchungen und 4 wechselnde Zinssätze können berücksichtigt werden. Die Daten werden in Tabellenform auf dem Bildschirm gelistet, Korrekturen der schon eingegebenen Daten sind möglich. Eine leichtverständliche Benutzerführung ermöglicht es auch dem Laien, dieses Programm anzuwenden. Im Dialog führt Sie der Computer durch das Programm und akzeptiert Eingaben für die einzelnen Teilprogramme über Tastaturabfrage, so daß eine nachträgliche Bestätigung durch –ENTER– entfällt.

Sollten Sie glücklicher Besitzer eines Ext.-Basicmoduls sein, wäre es sehr zu empfehlen, das Programm in diese Sprache umzuschreiben, da umfangreiche Berechnungen durchgeführt werden und die Rechenzeit dem TI-Basic entsprechend lang ist. Besitzen Sie eine Speichererweiterung, so brauchen nur die CHR$ (151) und CHR$ (152) geändert zu werden, ohne Speichererweiterung muß zusätzlich die Dimensionierung verkürzt werden.

Heinz Behrens

```
100 REM********************
110 REM*                  *
120 REM*   Jahreszinsen   *
130 REM*   Copyright by   *
140 REM*  Heinz  Behrens  *
150 REM*                  *
160 REM*  Bearbeitet vom  *
170 REM*    Team  des     *
180 REM* Aktuell Verlages *
190 REM*     Muenchen     *
200 REM*                  *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole *
230 REM*                  *
270 REM* Speicherbelegung *
280 REM*   10542  Bytes   *
290 REM********************
295 REM
300 REM  PROGRAMM ZUR BE-
         RECHNUNG DER ZINSEN
         (C) HEINZ BEHRENS
             2965 IHLOW.

310 REM       ** 1984 **
320 CN$="ja(j)/nein(n)"
330 CALL CHAR(150,"FFFF")
340 CALL CHAR(151,"010101010
1010101")
350 CALL CHAR(152,"808080808
080808")
360 GOSUB 3810
370 PRINT : : : : :"software
 behrens":"*1984*":TAB(23);"
>>A"
380 CALL SOUND(500,1000,0)
390 CALL KEY(1,KEY,STATUS)
400 IF KEY=1 THEN 410 ELSE 3
90
410 GOSUB 3810
420 PRINT : : : : : :"INFORM
ATIONEN ABFORDERN .I": :"ZIN
SRECHNUNG .............Z": :"
PROGRAMM ABBRECHEN ......A":
 :
430 CALL SOUND(500,1000,0)
440 CALL KEY(0,KEY,STATUS)
450 IF (KEY=90)+(KEY=122)THE
N 500
460 IF (KEY=65)+(KEY=97)THEN
 3100
470 IF (KEY=73)+(KEY=105)THE
N 480 ELSE 440
480 GOSUB 3440
490 GOTO 410
500 REM EINGABEPROGRAMM ZINS
SATZ
510 UI=78
520 CALL CLEAR
530 T(1)=1
540 M(1)=1
550 ZU=1
560 INPUT ".ZINSSATZ.......
.............AM 1.1. .......
..":Q(1)
570 P(1)=Q(1)/100
580 ZAN=0
590 PRINT : : :
600 PRINT "zinssatzaenderung
en ?........":CN$
610 CALL SOUND(200,1000,0)
620 CALL KEY(0,ZFR,S)
630 IF S=0 THEN 620
640 CALL HCHAR(23,29,ZFR)
650 IF (ZFR=78)+(ZFR=110)THE
N 880
660 IF (ZFR=74)+(ZFR=106)THE
N 670 ELSE 620
670 ZAN=ZAN+1
680 ZU=ZAN+1
690 CALL CLEAR
700 PRINT : : :ZAN;".";"ZINS
SATZAENDERUNG": : :" "
710 INPUT ".........DATUM:"
:DAT$
720 IF DAT$="" THEN 710
730 IF (ASC(DAT$)>47)+(ASC(D
AT$)<58)<>-2 THEN 710
740 GOSUB 3320
750 M(ZU)=MT
760 T(ZU)=TG
770 INPUT ".......ZINSSATZ:"
:Q(ZU)
780 P(ZU)=Q(ZU)/100
790 IF ZAN=3 THEN 880
800 PRINT : : :
810 PRINT "weitere.........
...........zinssatzaenderung
?":CN$
820 CALL SOUND(200,1000,0)
830 CALL KEY(0,ZFR,S)
840 IF S=0 THEN 830
850 CALL HCHAR(23,29,ZFR)
860 IF (ZFR=74)+(ZFR=106)THE
N 670
870 IF (ZFR=78)+(ZFR=110)THE
N 880 ELSE 830
880 TZ1$="..DATUM..!....ZINS
SATZ...."
890 FOR ZK=ZAN+2 TO 5
900 ZT(ZK)=360
910 P(ZK)=0
920 NEXT ZK
930 FOR ZY=2 TO ZAN+1
940 TAG=T(ZY)
950 MONAT=M(ZY)
960 GOSUB 3110
970 ZT(ZY)=F
980 NEXT ZY
990 GOSUB 3970
1000 REM TABELLE ZINSSATZ
1010 GOSUB 3890
1020 FOR ZN=1 TO ZU
```

```
1030 IF T(ZN)>=10 THEN 1040
ELSE 1060
1040 TB=2
1050 GOTO 1070
1060 TB=3
1070 IF M(ZN)>=10 THEN 1080
ELSE 1100
1080 TA=11
1090 GOTO 1110
1100 TA=10
1110 PRINT CHR$(151);".....
...!.............";CHR$(1
52)
1120 PRINT CHR$(151);".....
...!.............";CHR$(1
52):CHR$(151);" AB DEM..!...
.............";CHR$(152)
1130 PRINT CHR$(151);TAB(TB)
;T(ZN);TAB(6);M(ZN);TAB(11);
"!";TAB(16);P(ZN)*100;"%";TA
B(28);CHR$(152)
1140 CALL HCHAR(23,7,46)
1150 CALL HCHAR(23,TA,46)
1160 NEXT ZN
1170 PRINT : :
1180 PRINT "fehlerhafte eing
abe?":CN$
1190 CALL SOUND(200,1000,0)
1200 CALL KEY(0,ZFR,S)
1210 IF S=0 THEN 1200
1220 CALL HCHAR(23,29,ZFR)
1230 IF (ZFR=78)+(ZFR=110)TH
EN 1250
1240 IF (ZFR=74)+(ZFR=106)TH
EN 520 ELSE 1200
1250 CALL CLEAR
1260 REM EINGABE DER BUCHUNG
EN
1270 DIM E(100),DT(100),DM(1
00)
1280 N=1
1290 TY=9
1300 W$="M"
1310 V=78
1320 IF UI=74 THEN 1800
1330 PRINT "KONTOSTAND":"AM
31.12.":"DES VORJAHRES"
1340 INPUT "(VORTRAG) ? ....
..":E(1)
1350 DT(1)=1
1360 DM(1)=1
1370 CALL CLEAR
1380 PRINT ".....** ACHTUNG
**": :"..betraege die abgeho
ben":"..werden,muessen ein":
"..negatives vorzeichen"
1390 PRINT "..erhalten!": :"
.....** ": : :
1400 CALL HCHAR(24,3,150,28)
1410 PRINT : : : : : : : :"r
egelmaessige.....m
onatliche.....b
uchungen ?":CN$
1420 CALL SOUND(200,1000,0)
1430 CALL KEY(0,EFR,S)
1440 IF S=0 THEN 1430
1450 CALL HCHAR(23,29,EFR)
1460 IF (EFR=78)+(EFR=110)TH
EN 1630
1470 IF (EFR=74)+(EFR=106)TH
EN 1480 ELSE 1430
1480 PRINT : : : : : : : : :
 : :
1490 INPUT "DATUM DER.......
.....1.REGELM. ......
.....BUCHUNG ? ......
...":DAT$
1500 IF DAT$="" THEN 1490
1510 IF (ASC(DAT$)>47)+(ASC(
DAT$)<58)<>-2 THEN 1490
1520 GOSUB 3320
1530 RT=TG
1540 RM=MT
1550 PRINT
1560 INPUT "HOEHE DER,......
.....REGELM. ........
.....BUCHUNG ? ......
...":RE
1570 FOR K=2 TO 14-RM
1580 DT(K)=RT
1590 DM(K)=K+RM-2
1600 E(K)=RE
1610 NEXT K
1620 GOTO 1640
1630 K=2
1640 CALL CLEAR
1650 PRINT "UNREGELMAESSIGE
BUCHUNGEN...----------------
---------"
1660 PRINT "falls keine weit
eren.....buchungen :":"we
iter mit..enter'"
1670 PRINT : :
1680 INPUT "..DATUM:":DAT$
1690 IF DAT$="" THEN 1780
1700 IF (ASC(DAT$)>47)+(ASC(
DAT$)<58)<>-2 THEN 1680
1710 GOSUB 3320
1720 DT(K)=TG
1730 DM(K)=MT
1740 INPUT "BUCHUNG:":E(K)
1750 K=K+1
1760 IF K=100 THEN 2550
1770 GOTO 1670
1780 DM(K)=0
1790 E(K)=0
1800 TZ1$="..DATUM..!.....BU
CHUNG...."
1810 IF N<=1 THEN 1820 ELSE
1870
```

```
1820 N=2
1830 GOSUB 3890
1840 PRINT CHR$(151);"......
...|.................";CHR$(1
52)
1850 PRINT CHR$(151);" VORTR
AG";TAB(11);"|";TAB(12);E(1)
;"DM";TAB(28);CHR$(152)
1860 IF K=2 THEN 2060 ELSE 1
880
1870 GOSUB 3890
1880 PRINT CHR$(151);"......
...|.................";CHR$(1
52)
1890 IF DT(N)>=10 THEN 1900
ELSE 1920
1900 TB=2
1910 GOTO 1930
1920 TB=3
1930 IF DM(N)>=10 THEN 1940
ELSE 1960
1940 TA=11
1950 GOTO 1970
1960 TA=10
1970 PRINT CHR$(151);TAB(TB)
;DT(N);TAB(6);DM(N);TAB(11);
"|";TAB(12);E(N);"DM";TAB(28
);CHR$(152)
1980 CALL HCHAR(23,7,46)
1990 CALL HCHAR(23,TA,46)
2000 N=N+1
2010 IF (N=TY)+(N=TY+8)=-1 T
HEN 2070
2020 IF (N=TY+16)+(N=TY+24)+
(N=TY+32)+(N=TY+40)=-1 THEN
2070
2030 IF (N=TY+48)+(N=TY+56)+
(N=TY+64)+(N=TY+72)=-1 THEN
2070
2040 IF (N=TY+80)+(N=TY+88)+
(N=TY+94)=-1 THEN 2070
2050 IF N<>K THEN 1880
2060 TY=N
2070 IF W$="P" THEN 2200
2080 PRINT "................
............korrekturen erfo
rderlich?":CN$
2090 CALL SOUND(200,1000,0)
2100 CALL KEY(0,EFR,S)
2110 IF S=0 THEN 2100
2120 CALL HCHAR(23,29,EFR)
2130 IF (EFR=78)+(EFR=110)TH
EN 2180
2140 IF (EFR=74)+(EFR=106)TH
EN 2150 ELSE 2100
2150 W$="P"
2160 N=N-8
2170 GOTO 1810
2180 W$="M"
2190 IF N=K THEN 2580 ELSE 1
810
2200 PRINT " ":"EINGABE DER
FEHLERHAFTEN....DATEN":" "
2210 W$="M"
2220 INPUT " DATUM:":DAT$
2230 IF (DAT$="VORTRAG")+(DA
T$="vortrag")THEN 2300
2240 IF DAT$="" THEN 2220
2250 IF (ASC(DAT$)>47)+(ASC(
DAT$)<58)<>-2 THEN 2220
2260 GOSUB 3320
2270 FT=TG
2280 FM=MT
2290 INPUT "BETRAG:":FE
2300 PRINT " ": :"EINGABE DE
R RICHTIGEN DATEN": :" "
2310 IF (DAT$<>"VORTRAG")+(D
AT$<>"vortrag")=0 THEN 2340
2320 PRINT " DATUM:VORTRAG"
2330 GOTO 2400
2340 INPUT " DATUM:":DAT$
2350 IF DAT$="" THEN 2340
2360 IF (ASC(DAT$)>47)+(ASC(
DAT$)<58)<>-2 THEN 2340
2370 GOSUB 3320
2380 KT=TG
2390 KM=MT
2400 INPUT "BETRAG:":KE
2410 IF (DAT$="VORTRAG")+(DA
T$="vortrag")THEN 2420 ELSE
2450
2420 E(I)=KE
2430 N=N-8
2440 GOTO 1810
2450 FOR I=1 TO K-1
2460 IF (DT(I)=FT)+(DM(I)=FM
)+(E(I)=FE)=-3 THEN 2470 ELS
E 2520
2470 DT(I)=KT
2480 DM(I)=KM
2490 E(I)=KE
2500 N=N-8
2510 GOTO 1810
2520 NEXT I
2530 N=N-8
2540 GOTO 1810
2550 CALL CLEAR
2560 PRINT "DIE ANZAHL DER B
UCHUNGEN....BETRAGT 100.MEHR
 BUCHUNGEN..KOENNEN NICHT VE
RARBEITET...WERDEN"
2570 GOTO 3030
2580 CALL CLEAR
2590 PRINT "sind buchungen v
ergessen....worden?":CN$
2600 CALL SOUND(200,1000,0)
2610 CALL KEY(0,V,S)
2620 IF S=0 THEN 2610
2630 CALL HCHAR(23,29,V)
2640 IF (V=78)+(V=110)THEN 2
660
```

```
2650 IF (V=74)+(V=106)THEN 1
640 ELSE 2610
2660 REM BERECHNUNG
2670 KAP=E(1)
2680 X=0
2690 GOSUB 3160
2700 XA=E(1)
2710 ZIN=0
2720 FOR O=1 TO ZU
2730 ZIN=ZIN+XA*P(O)*RX(O)/3
60
2740 RD=ZIN
2750 GOSUB 3260
2760 ZIN=RD
2770 NEXT O
2780 FOR G=2 TO K-1
2790 XA=E(G)
2800 IF XA>0 THEN 2830
2810 TAG=DT(G)
2820 GOTO 2840
2830 TAG=DT(G)+1
2840 MONAT=DM(G)
2850 GOSUB 3110
2860 X=F
2870 GOSUB 3160
2880 FOR O=1 TO ZU
2890 ZIN=ZIN+XA*P(O)*RX(O)/3
60
2900 RD=ZIN
2910 GOSUB 3260
2920 ZIN=RD
2930 NEXT O
2940 KAP=KAP+E(G)
2950 NEXT G
2960 CALL CLEAR
2970 CALL HCHAR(21,3,42,29)
2980 PRINT "..KONTOSTAND":".
.AM 31.12. .";KAP;"DM": :"..
ZINSEN ....";ZIN;"DM": :"  K
ONTOSTAND"
2990 PRINT "..AM 31.12.":"..
+ ZINSEN ..";KAP+ZIN;"DM": :
 : : : : :
3000 CALL HCHAR(20,3,42,29)
3010 CALL VCHAR(8,3,42,13)
3020 CALL VCHAR(8,31,42,13)
3030 PRINT "daten nochmals b
e-.........trachten bzw.kor
-.........rigieren?":CN$
3040 CALL SOUND(200,1000,0)
3050 CALL KEY(0,UI,S)
3060 IF S=0 THEN 3050
3070 CALL HCHAR(23,29,UI)
3080 IF (UI=74)+(UI=106)THEN
 880
3090 IF (UI=78)+(UI=110)THEN
 410 ELSE 3050
3100 END
3110 REM UP ZUR BERECHNUNG D
ER TAGE IM JAHR UNTER EINGAB
E DES DATUMS
3120 F=30*(MONAT-1)+TAG-1
3130 IF TAG=32 THEN 3140 ELS
E 3150
3140 F=F-1
3150 RETURN
3160 REM RASTER ZINSSATZAEND
ERUNGEN
3170 FOR RN=2 TO 5
3180 IF (X<ZT(RN))+(X>=ZT(RN
-1))=-2 THEN 3210
3190 RX(RN-1)=0
3200 NEXT RN
3210 RX(RN-1)=ZT(RN)-X
3220 FOR RK=RN+1 TO 5
3230 RX(RK-1)=ZT(RK)-ZT(RK-1
)
3240 NEXT RK
3250 RETURN
3260 REM UP RUNDEN
3270 IF RD>=INT(100*RD)/100+
.005 THEN 3300
3280 RD=INT(100*RD)/100
3290 RETURN
3300 RD=INT(100*RD)/100+.01
3310 RETURN
3320 REM UP DATUM
3330 IF (POS(DAT$,".",4))<>0
 THEN 3340 ELSE 3350
3340 DAT$=SEG$(DAT$,1,(POS(D
AT$,".",4)-1))
3350 TG=INT(VAL(DAT$))
3360 ST=LEN(DAT$)-LEN(STR$(T
G))
3370 IF ASC(DAT$)=48 THEN 33
80 ELSE 3390
3380 ST=ST-1
3390 IF ST=3 THEN 3400 ELSE
3420
3400 MT=(VAL(DAT$)-TG)*100
3410 RETURN
3420 MT=(VAL(DAT$)-TG)*10
3430 RETURN
3440 REM UP INFO
3450 CALL CLEAR
3460 PRINT ".........** INFO
 **": :
3470 PRINT "mit diesem progr
amm koennen die jaehrlichen
zinsen eineskontos berechnet
 werden. "
3480 PRINT "das programm ver
arbeitet biszu 100 buchungen
 und 4 ver- schiedene zinssa
etze."
3490 PRINT "die buchungen un
d zinssaetzemuessen im zeitr
aum vom 1.1.bis 31.12. eines
 jahres":"liegen."
3500 PRINT "bei abbuchungen
wird ange-..nommen, dass die
```

```
 betraege...rechtzeitig geku
endigt.."
3510 PRINT "wurden.-bei spar
einlagen mitgesetzlicher kue
ndigungs-...frist koennen bi
s zu 2000 dm"
3520 PRINT "ohne kuendigung
innerhalb...von 4 wochen abg
ehoben.....werden. anderfal
ls werden "
3530 PRINT "vorschusszinsen
faellig."
3540 PRINT TAB(25);">>A"
3550 CALL SOUND(500,1000,0)
3560 CALL KEY(1,KEY,STATUS)
3570 IF KEY=1 THEN 3580 ELSE
 3560
3580 CALL CLEAR
3590 PRINT "weil die banken
z.t. von-...einander abweich
ende be-....rechnungs- und r
undungs- "
3600 PRINT "vorschriften anw
enden,.....koennen geringfu
egige ab-...weichungen auftr
eten."
3610 PRINT TAB(25);">>A"
3620 CALL SOUND(500,1000,0)
3630 CALL KEY(1,KEY,STATUS)
3640 IF KEY=1 THEN 3650 ELSE
 3630
3650 CALL CLEAR
3660 PRINT "eingabebeispiel
zinssatz...----------------
---------":"(zinssatz in pro
zent)": : :
3670 PRINT "...DATUM: 3.5. .
..........ZINSSATZ: 4": :T
AB(25);">>A"
3680 CALL SOUND(500,1000,0)
3690 CALL KEY(1,KEY,STATUS)
3700 IF KEY=1 THEN 3710 ELSE
 3690
3710 CALL CLEAR
3720 PRINT "eingabebeispiel
buchung":"----------------
-----":"(buchung in dm)": :
:
3730 PRINT "...DATUM: 7.5. .
............BUCHUNG: 1201.2
3": :
3740 PRINT : :"betraege,die
abgehoben.....werden, muess
en ein.....negatives vor
zeichen er-":"halten.": : :
3750 PRINT "sollen keine buc
hungen mehr eingegeben werde
n, ist die..taste 'enter' zu
 betaetigen.": : :TAB(25);">
>A"
3760 CALL SOUND(500,1000,0)
3770 CALL KEY(1,KEY,STATUS)
3780 IF KEY=1 THEN 3790 ELSE
 3770
3790 CALL CLEAR
3800 RETURN
3810 REM UP KOPF
3820 CALL CLEAR
3830 PRINT "..PROGRAMM ZUR B
ERECHNUNG.....DER ZINS
EN": : : :
3840 CALL HCHAR(17,4,42,26)
3850 CALL HCHAR(22,4,42,26)
3860 CALL VCHAR(17,3,42,6)
3870 CALL VCHAR(18,29,42,4)
3880 RETURN
3890 REM UP TABELLE
3900 CALL CLEAR
3910 CALL HCHAR(23,4,150,26)
3920 CALL HCHAR(23,3,151)
3930 CALL HCHAR(23,30,152)
3940 PRINT CHR$(151);TZ1$;CH
R$(152)
3950 PRINT CHR$(151);"------
---|----------------";CHR$(1
52)
3960 RETURN
3970 REM SORTIEREN DREIER ZA
HLEN
3980 SI=2
3990 SN=SI+1
4000 IF ZT(SI)<=ZT(SN)THEN 4
170
4010 GR=ZT(SI)
4020 KL=ZT(SN)
4030 SGR=P(SI)
4040 SKL=P(SN)
4050 MGR=M(SI)
4060 MKL=M(SN)
4070 TGR=T(SI)
4080 TKL=T(SN)
4090 ZT(SN)=GR
4100 ZT(SI)=KL
4110 P(SI)=SKL
4120 P(SN)=SGR
4130 M(SI)=MKL
4140 M(SN)=MGR
4150 T(SI)=TKL
4160 T(SN)=TGR
4170 IF SN=4 THEN 4200
4180 SN=SN+1
4190 GOTO 4000
4200 IF SI=3 THEN 4230
4210 SI=SI+1
4220 GOTO 3990
4230 RETURN
```

KONTOKORRENT–RECHNUNG

Das Programm berechnet nach der banküblichen Staffelmethode für den Zeitraum eines Quartals die aufgelaufenen Soll- und Habenzinsen, die Kontoführungs- und Postengebühren unter Berücksichtigung der eingegebenen Freiposten. Zuletzt wird noch die Umsatzprovision hinzugeschlagen und das Ganze mit Anfangs- und Abschlußsaldo auf dem Bildschirm ausgegeben. Zum Schluß noch ein Hinweis zur Anwendung dieses Programmes: Eingehende Geldbeträge werden ohne Vorzeichen eingegeben, abgehobene Beträge geben Sie bitte mit einem Minuszeichen (vorangestellt) ein.

Ludwig Blencke

TEXT-EDITOR

Text-Editor ist ein sehr komfortables Textprogramm mit einer Vielzahl von Funktionen. Sie können damit Texte erstellen, deren Länge nur von dem Ihnen zur Verfügung stehenden Speicherplatz abhängig ist. Durch eine integrierte Chardefinition kann der Text illustriert werden oder Sie können sich Sonderzeichen erstellen, sogar als reines Zeichenprogramm ist der Text-Editor anwendbar. Besitzer eines Druckers werden sich sicher nicht scheuen, ein paar Zeilen für die Druckerausgabe einzufügen.

BEDIENUNGSANLEITUNG

1. TEXT EINGEBEN
Nachdem eine Überschrift gewählt wurde, erscheint der Cursor in der linken oberen Bildschirmecke. Er ist über die Pfeiltasten in alle Richtungen steuerbar. Ein Piepston macht auf das Ende einer Zeile aufmerksam. Zum Weiterschreiben wird der Cursor mit FCTN 'X' in die nächste Zeile befördert. Ist eine Bildschirmseite vollgeschrieben, wird die ENTER-Taste gedrückt. Nun haben Sie zwei Möglichkeiten: Ist der Text beendet, führt ein nochmaliges Betätigen der ENTER-Taste ins Hauptmenü zurück. Möchten Sie am gleichen Text noch weiterschreiben, ist eine neue Überschrift für die zweite Bildschirmseite einzugeben. Später können die einzelnen Bildschirmseiten unter einer Überschrift zusammengefaßt werden.
2. SUCHEN NACH ÜBERSCHRIFTEN
Geben Sie die Überschrift des Textes ein, den Sie suchen. Findet der Computer keinen zugehörigen Text, können Sie es nach einer Fehlermeldung mit einer anderen Überschrift noch einmal versuchen. Wird der passende Text jedoch gefunden, erscheint er nicht auf dem Bildschirm, sondern die Überschrift wird nur zwischen Begrenzungspfeile gesetzt. In dieser Phase können die Bildschirmseiten unter einer Überschrift zusammengefaßt werden. Ändern Sie die Überschriften aller Bildschirmseiten größer als 1 in die Überschrift der Bildschirmseite 1 um, wird der gesamte Text in der gleichen Reihenfolge unter dieser Überschrift eingebunden. Wählen Sie eine Überschrift, die noch nicht vorhanden ist, so wird die alte Überschrift durch diese ersetzt. Mit der ENTER-Taste kann der Text nun gelesen oder editiert werden. (Cursor frei steuerbar). Danach führt die ENTER-Taste ins Hauptmenü zurück.
3. TEXTSTELLE SUCHEN
Geben Sie ein Stichwort des Textes oder der Textstelle ein, die Sie suchen. Ist das Stichwort in einem der gespeicherten Text vorhanden, wird dieser ausgegeben. Der Cursor markiert die Zeile, in der das Stichwort vorhanden ist. Ein Editieren des Textes ist jetzt, wie schon vorher beschrieben, möglich. Nach dem Drücken der ENTER-Taste fragt der Computer, ob er noch einen anderen Text heraussuchen soll, in dem das Stichwort vorhanden ist. Geben Sie ein 'J' ein, wiederholt sich der Suchvorgang, bei Eingabe von 'N' bekommen Sie wieder das Hauptmenü.
ein, wiederholt sich der Suchvorgang, bei Eingabe von 'N' bekommen Sie wieder das Hauptmenü.
4. TEXTE EINLESEN
Diese Funktion liest auf Kassettenrekorder gespeicherte Texte in den Speicher ein.
5. TEXTE ABSPEICHERN
Ihre Daten werden auf Kassette abgespeichert.
6. TEXTE LÖSCHEN
Die gesamten im Speicher befindlichen Texte werden gelöscht.
7. BASIC
Löscht den Bildschirm und beendet das Programm. Befinden Sie sich im Menüteil 1, stehen Ihnen folgende Sonderfunktionen zur Verfügung:
1. BILDSCHIRMFARBE
Während der Eingabe Ihres Textes können Sie unter verschiedenen Bildschirmfarben wählen. Die folgenden Befehle geben Sie am Anfang einer beliebigen Zeile ein. Es spielt dabei keine Rolle, ob die Zeile beschrieben ist oder nicht. Nach Ausführung des Befehls wird die Zeile in ihrer ursprünglichen Form wieder hergestellt.

".C0" = schwarze Schrift auf kornblumenblaumem Hintergrund mit grünen Streifen.
".Co" = weiße Schrift auf dunkelblauem Hintergrund.

2. ZEILENABSTAND

".L0" = einzeilig
".L1" = zweizeilig

3. TEXT VERSCHIEBEN

".V-" = löscht die Zeile, in der der Cursor steht und rückt den nachfolgenden Text jeweils eine Zeile hoch.
".V+" = verschiebt den Text um jeweils eine Zeile nach unten und löscht Zeile 24. Mit dieser Funktion ist es möglich, Zeilen in den Text einzuschieben.

4. SONDERZEICHEN
"."& ASCII von 032-143 & Space & Hexadezimalcode. Generiert das von Ihnen gewünschte Sonderzeichen wie im folgenden Beispiel:
".042 AA55AA55AA55AA55"

5. RÜCKSETZEN DER BILDSCHIRMFARBE UND DER SONDERZEICHEN.

".RTN" = stellt den ASCII-Code und die ursprüngliche Bildschirmfarbe wieder her. Achtung: Nur die Zeichen 32–95 werden zurückgesetzt.

Alle Sonderfunktionen werden ohne Anführungsstriche eingegeben, sie dienen in der Beschreibung nur der besseren Kenntlichmachung.

Thomas Götz

```
100 ! **********************
110 ! *                    *
120 ! *KONTOKORRENTRECHNUNG*
130 ! *    Copyright by    *
140 ! *   Ludwig Blencke   *
150 ! *                    *
160 ! *   Bearbeitet vom   *
170 ! *      Team des      *
180 ! *  Aktuell Verlages  *
190 ! *      Muenchen      *
200 ! *                    *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole  *
230 ! *     Ext. Basic     *
240 ! *                    *
270 ! *  Speicherbelegung  *
280 ! *     3895 Bytes     *
290 ! **********************
295 !
300 CALL CLEAR :: DISPLAY AT
(1,5):"KONTOKORRENTRECHNUNG"
:"....FUER GEHALTSKONTEN"
310 DISPLAY AT(4,1):"QUARTAL
(1-4):..JAHR 19"
315 ON BREAK NEXT :: ON WARN
ING NEXT
320 OPTION BASE 1
330 DIM T(30,3)
340 ACCEPT AT(4,14)VALIDATE(
"1234")SIZE(1)BEEP:Q :: ACCE
PT AT(4,23)VALIDATE(DIGIT)SI
ZE(2)BEEP:JA
350 IF Q<3 THEN QD$="31." EL
SE QD$="30."
360 IF Q=1 THEN QM=12 :: JA=
JA-1 ELSE QM=(Q-1)*3
370 IF QM=12 THEN QM$="12" E
LSE QM$="0"&STR$(QM)
380 QM$=QM$&".19" :: DISPLAY
 AT(6,1):"KONTOSTAND AM "&QD
$&QM$&STR$(JA):: DISPLAY AT(
7,17):USING "#######.## DM";A
S
390 ACCEPT AT(7,17)VALIDATE(
NUMERIC)SIZE(-9)BEEP:AS
400 FOR I=1 TO 30 :: T(I,1),
T(I,2),T(I,3)=0 :: NEXT I
410 DISPLAY AT(9,1):USING "G
RUNDGEBUEHR/MONAT=..#.## DM"
:GB :: ACCEPT AT(9,22)VALIDA
TE(NUMERIC)SIZE(-4)BEEP:GB
420 DISPLAY AT(11,1):USING "
POSTENGEBUEHR/BUCHG.= .## DM
":GP :: ACCEPT AT(11,23)VALI
DATE(NUMERIC)SIZE(-4)BEEP:GP
430 DISPLAY AT(13,1):USING "
FREIPOSTEN/MONAT=.....##":F
P :: ACCEPT AT(13,25)VALIDAT
E(NUMERIC)SIZE(-2)BEEP:FP
440 GOSUB 960 :: PZ,FP1,FP2,
FP3=0
450 GOSUB 930 :: DISPLAY AT(
9,1):"EINGABE:(BEENDEN MIT '
0')":"WERTSTELLUNG....BETRAG
 DM"
460 DISPLAY AT(11,1):"TAG..M
ONAT.......BETRAG":".....(1-
12).......(SOLL=-)" :: I=0
470 ACCEPT AT(13+I,1)VALIDAT
E(DIGIT)SIZE(2)BEEP:TG :: IF
 TG=0 THEN 600 :: ACCEPT AT(
13+I,6)VALIDATE(DIGIT)SIZE(2
)BEEP:MT
480 ACCEPT AT(13+I,18)VALIDA
TE(NUMERIC)BEEP:DM
490 MT=MT-(Q-1)*3 :: IF TG=3
1 THEN TG=30
500 IF MT<1 OR MT>3 THEN DIS
PLAY AT(13+I,1):"MONAT IM FA
LSCHEN QUARTAL!" ELSE GOTO 5
20
510 TG,MT,DM=0 :: GOTO 590
520 T(TG,MT)=T(TG,MT)+DM ::
PZ=PZ+1
530 ON MT GOTO 540,560,580
540 FP1=FP1+1 :: IF FP1>=FP
THEN FP1=FP
550 GOTO 590
560 FP2=FP2+1 :: IF FP2>=FP
THEN FP2=FP
570 GOTO 590
580 FP3=FP3+1 :: IF FP3>=FP
THEN FP3=FP
590 I=I+1 :: IF I>10 THEN 45
0 ELSE 470
600 SO,HA,DM,TG1,M=0 :: FP=F
P1+FP2+FP3 :: FOR J=1 TO 3 :
: FOR I=1 TO 30
610 IF J=MTZ AND I=TGZ AND M
>-1 THEN 650
620 IF J=MTZ AND I=TGZ THEN
640
630 IF T(I,J)=0 THEN 690
640 IF M THEN 670 ELSE M=-1
650 DM=AS :: TG=I+(J-1)*30-T
G1 :: GOSUB 900
660 DM=AS+T(I,J):: GOTO 680
670 TG=I+(J-1)*30-TG1 :: GOS
UB 900 :: DM=DM+T(I,J)
680 TG1=TG1+TG :: IF J=MTZ A
ND I=TGZ AND(J<3 OR(J=3 AND
TGZ<30))THEN GOSUB 1080 :: G
OSUB 960
```

```
690 NEXT I
700 NEXT J
710 GOSUB 1080 :: GOSUB 930
720 IF Q=1 OR Q=4 THEN QD$="
31." ELSE QD$="30."
730 IF Q=1 THEN JA=JA+1
740 DISPLAY AT(9,1):"ABSCHLU
SS PER "&QD$&STR$(Q*3)&".19"
&STR$(JA)
750 DISPLAY AT(12,1):USING "
ROHSALDO=.......######.## DM
":DM
760 DISPLAY AT(13,1):USING "
SOLLZINSEN=.....######.## DM
":SOZ
770 DISPLAY AT(14,1):USING "
HABENZINSEN=....######.## DM
":HAZ
780 DISPLAY AT(15,1):"UMSATZ
PROVISION=...%" :: ACCEPT AT
(15,17)VALIDATE(NUMERIC)SIZE
(-3)BEEP:UP
790 UP=INT(SOU*UP/360*100+.5
)/100
800 DISPLAY AT(16,17):USING
"######.## DM":UP
810 GB=GB*3+(PZ-FP)*GP :: DI
SPLAY AT(17,1):USING "GEBUEH
REN=......######.## DM":GB
820 IF Q=4 THEN JA=JA+1
830 CALL HCHAR(18,1,95,32)::
 DISPLAY AT(19,1):"SALDO PER
 01."&STR$(Q*3+1)&".19"&STR$
(JA)
840 DISPLAY AT(20,16):USING
"=######.## DM":DM+HAZ-SOZ-U
P-GB :: CALL HCHAR(21,1,61,3
2)
850 DISPLAY AT(22,1):USING "
BUCHUNGEN INSGESAMT=...##":P
Z
860 DISPLAY AT(24,1):"WEITER
E BERECHNUNGEN(J/N)?" :: ACC
EPT AT(24,22)VALIDATE("JNjn"
)SIZE(-1)BEEP:Q$
870 I,J,DM,SOZ,HAZ,SOU,Q,JA,
AS,GB,GP,FP,TGA,MTA,TGZ,MTZ,
UP=0
880 IF Q$="J" OR Q$="j" THEN
 300
890 CALL CLEAR :: END
900 ! ZINSZAHLENBERECHNUNG *
*****************************
*****************************
910 IF SGN(DM)=-1 THEN SO=SO
+INT(ABS(DM)/100+.5)*TG ELSE
 HA=HA+INT(DM/100+.5)*TG
920 RETURN
930 !   TEILLOESCHUNG
      BILDSCHIRM          *
*****************************
*****************************
940 CALL HCHAR(9,1,32,480)
950 RETURN
960 !   ZINSZEITRAEUME
      EINGEBEN            *
*****************************
*****************************
970 GOSUB 930
980 DISPLAY AT(9,1):"ZINSZEI
TRAUM EINGEBEN":"VON TAG/MON
AT BIS TAG/MONAT"
990 ACCEPT AT(11,5)VALIDATE(
DIGIT)SIZE(2)BEEP:TGA :: ACC
EPT AT(11,9)VALIDATE(DIGIT)S
IZE(2)BEEP:MTA :: IF TGA=31
THEN TGA=30
1000 IF MTZ+(Q-1)*3>MTA THEN
 1030
1010 ACCEPT AT(11,19)VALIDAT
E(DIGIT)SIZE(2)BEEP:TGZ :: A
CCEPT AT(11,23)VALIDATE(DIGI
T)SIZE(2)BEEP:MTZ :: IF TGZ=
31 THEN TGZ=30
1020 IF TGZ<=I AND MTZ<=J OR
 MTZ<MTA OR TGA<1 OR TGA>30
OR TGZ<1 OR TGZ>30 OR(MTA=MT
Z AND TGA>TGZ)THEN 1030 ELSE
 1040
1030 MTZ=0 :: DISPLAY AT(13,
1):"FEHLER,EINGABE WIEDERHOL
EN!" :: GOTO 990
1040 MTZ=MTZ-(Q-1)*3 :: IF M
TZ>3 THEN MTZ=3 :: TGZ=30
1050 DISPLAY AT(13,1):"EINGA
BE SOLL- / HABENZINSEN" :: D
ISPLAY AT(14,26):"%" :: ACCE
PT AT(14,9)VALIDATE(NUMERIC)
SIZE(-5)BEEP:SZ
1060 ACCEPT AT(14,17)VALIDAT
E(NUMERIC)SIZE(-5)BEEP:HZ ::
 GOSUB 930
1070 RETURN
1080 !   ZINSBERECHNUNG
*****************************
*****************************
1090 SOZ=SOZ+INT(SO*SZ/360*1
00+.5)/100
1100 HAZ=HAZ+INT(HA*HZ/360*1
00+.5)/100
1110 SOU=SOU+SO :: SO,HA=0
1120 RETURN
```

```
100 ! ************************
110 ! *                      *
120 ! *      TEXTEDITOR      *
130 ! *    Copyright by      *
140 ! *    Thomas Goetz      *
150 ! *                      *
160 ! *   Bearbeitet vom     *
170 ! *       Team des       *
180 ! *  Aktuell Verlages    *
190 ! *       Muenchen       *
200 ! *                      *
210 ! * Benoetigte Geraete   *
220 ! *  TI99/4A  Konsole    *
230 ! *      Ext. Basic      *
240 ! * Kassettenrekorder    *
250 ! *                      *
270 ! *  Speicherbelegung    *
280 ! *      3471 Bytes      *
290 ! ************************
295 !
300 DIM T$(12,24),U$(12):: T
N$,TT$="" :: K,S,L,I=0 :: M=
1 :: P=-32651
310 CALL CLEAR :: ON BREAK N
EXT
320 DISPLAY AT(3,1):"...***.
.UEBERSICHT..***"
330 DISPLAY AT(8,1):"ÜTEXT S
CHREIBEN.....A1Ü"
340 DISPLAY AT(10,1):"ÜSUCHE
N NACH STICHWORT...A2Ü"
350 DISPLAY AT(12,1):"ÜSUCHE
N NACH TEXTWORT....A3Ü"
360 DISPLAY AT(14,1):"ÜTEXTE
 EINLESEN.....A4Ü"
370 DISPLAY AT(16,1):"ÜTEXTE
 ABSPEICHERN.....A5Ü"
380 DISPLAY AT(18,1):"ÜTEXTE
 LOESCHEN.....A6Ü"
390 DISPLAY AT(20,1):"ÜBASIC
.....A7Ü"
400 DISPLAY AT(24,1):"WUNSCH
NUMMER EINGEBEN:"
410 CALL KEY(0,K,S):: IF S=0
 THEN 410 :: CALL HCHAR(24,2
6,K):: IF K<49 OR K>55 THEN
410
420 ON K-48 GOTO 450,590,690
,780,870,430,440
430 RUN
440 CALL CLEAR :: END
450 CALL CLEAR
460 DISPLAY AT(1,1):"STICHWO
RT:"
470 DISPLAY AT(2,1):">.....
.....<"
480 ACCEPT BEEP AT(2,2)SIZE(
-20):TN$ :: IF TN$="" THEN 3
10
490 FOR I=1 TO 12 :: IF U$(I
)="" THEN 520 ELSE IF U$(I)<
>TN$ THEN 510
500 FOR L=1 TO 24 :: DISPLAY
 AT(L,1):T$(I,L):: NEXT L ::
 L=1 :: GOTO 530
510 NEXT I
520 U$(I)=TN$ :: L=1 :: CALL
 CLEAR
530 GOSUB 540 :: GOTO 450
540 ACCEPT BEEP AT(L,1)SIZE(
-28):TT$ :: CALL PEEK(P,K)::
 IF K=13 AND SEG$(TT$,1,1)="
." THEN 960
550 T$(I,L)=TT$
560 IF K=11 THEN L=L-M :: IF
 L<1 THEN L=25-M
570 IF K=10 THEN L=L+M :: IF
 L>24 THEN L=1
580 IF K=13 THEN RETURN ELSE
 540
590 CALL CLEAR :: S=0
600 DISPLAY AT(1,1):"STICHWO
RT:"
610 ACCEPT BEEP AT(2,1)SIZE(
-20):TN$
620 IF TN$="" THEN 310
630 FOR I=1 TO 12 :: IF U$(I
)<>TN$ THEN 640 ELSE 660
640 NEXT I :: IF S=1 THEN 59
0
650 DISPLAY AT(24,1):"STICHW
ORT NICHT GEFUNDEN!" :: FOR
L=1 TO 200 :: NEXT L :: CALL
 HCHAR(24,1,32,32):: GOTO 61
0
660 IF S=0 THEN DISPLAY AT(2
,1):">"&U$(I);TAB(22);"<" ::
 ACCEPT BEEP AT(2,2)SIZE(-20
):U$(I):: IF U$(I)="" THEN 6
60
670 FOR L=1 TO 24 :: DISPLAY
 AT(L,1):T$(I,L):: NEXT L
680 L=1 :: GOSUB 540 :: CALL
 CLEAR :: S=1 :: GOTO 640
690 CALL CLEAR
700 DISPLAY AT(1,1):"TEXTWOR
T:"
710 ACCEPT BEEP AT(2,1)SIZE(
-28):TN$ :: IF TN$="" THEN 3
10
720 FOR I=1 TO 12 :: FOR L=1
 TO 24
```

```
730 IF POS(T$(I,L),TN$,1)=0
THEN 770
740 CALL CLEAR :: FOR S=1 TO
 24 :: DISPLAY AT(S,1):T$(I,
S):: NEXT S :: GOSUB 540
750 CALL CLEAR :: DISPLAY AT
(1,1)BEEP:"WEITER? (J/N)"
760 CALL KEY(0,K,S):: IF S<1
 THEN 760 ELSE IF K=78 OR K=
106 THEN 690 ELSE CALL CLEAR
 :: GOTO 310
770 NEXT L :: NEXT I :: GOTO
 690
780 ON ERROR 860
790 OPEN #1:"CS1",INTERNAL,I
NPUT ,FIXED 192
800 INPUT #1:S
810 FOR K=1 TO S
820 INPUT #1:U$(K),T$(K,1),T
$(K,2),T$(K,3),T$(K,4),T$(K,
5),T$(K,6)
830 INPUT #1:T$(K,7),T$(K,8)
,T$(K,9),T$(K,10),T$(K,11),T
$(K,12)
840 INPUT #1:T$(K,13),T$(K,1
4),T$(K,15),T$(K,16),T$(K,17
),T$(K,18):: INPUT #1:T$(K,1
9),T$(K,20),T$(K,21),T$(K,22
),T$(K,23),T$(K,24)
850 NEXT K
860 ON ERROR 310 :: CLOSE #1
 :: GOTO 310
870 OPEN #1:"CS1",INTERNAL,O
UTPUT,FIXED 192
880 ON ERROR 310
890 FOR S=1 TO 12 :: IF U$(S
)="" THEN PRINT #1:S-1 :: GO
TO 910
900 NEXT S
910 FOR K=1 TO S-1
920 PRINT #1:U$(K),T$(K,1),T
$(K,2),T$(K,3),T$(K,4),T$(K,
5),T$(K,6)
930 PRINT #1:T$(K,7),T$(K,8)
,T$(K,9),T$(K,10),T$(K,11),T
$(K,12)
940 PRINT #1:T$(K,13),T$(K,1
4),T$(K,15),T$(K,16),T$(K,17
),T$(K,18):: PRINT #1:T$(K,1
9),T$(K,20),T$(K,21),T$(K,22
),T$(K,23),T$(K,24)
950 NEXT K :: CLOSE #1 :: GO
TO 310
960 IF POS(TT$,".L1",1)THEN
M=2 :: GOTO 1040 :: ELSE IF
POS(TT$,".LO",1)THEN M=1 ::
GOTO 1040
970 IF POS(TT$,".CO",1)THEN
CALL SCREEN(13):: CALL COLOR
(1,2,8,2,2,8,3,2,8,4,2,8,5,2
,8,6,2,8,7,2,8,8,2,8):: GOTO
 1040
980 IF POS(TT$,".V+",1)THEN
1050 ELSE IF POS(TT$,".V-",1
)THEN 1060
990 IF POS(TT$,".C1",1)THEN
CALL SCREEN(5):: CALL COLOR(
1,16,1,2,16,1,3,16,1,4,16,1,
5,16,1,6,16,1,7,16,1,8,16,1)
:: GOTO 1040
1000 IF POS(TT$,".RTN",1)THE
N CALL CHARSET :: GOTO 1040
1010 ON ERROR 1040
1020 IF ASC(SEG$(TT$,2,1))>5
7 OR LEN(TT$)<6 OR ASC(SEG$(
TT$,5,1))<>32 THEN 1040
1030 TT$=SEG$(TT$,2,27):: CA
LL CHAR(VAL(SEG$(TT$,1,3)),S
EG$(TT$,5,16))
1040 DISPLAY AT(L,1):T$(I,L)
:: GOTO 540
1050 FOR S=23 TO L STEP -1 :
: T$(I,S+1)=T$(I,S):: DISPLA
Y AT(S+1,1):T$(I,S+1):: NEXT
 S :: T$(I,L)="" :: GOTO 104
0
1060 FOR S=L TO 23 :: T$(I,S
)=T$(I,S+1):: DISPLAY AT(S,1
):T$(I,S):: NEXT S :: T$(I,2
4)="" :: DISPLAY AT(24,1)::
GOTO 540
```

SPIDER GAME

Das Programm SPIDER GAME wurde auf einem TI 99/4A in der Sprache Extended Basic geschrieben. Zur Steuerung des Programms ist eigentlich ein Joystick nötig; mit einer kleinen Hilfsroutine ist es jedoch möglich, SPIDER GAME über die Tastatur zu spielen. Die Bedienung erfolgt dann mit den Cursortasten (E, S, D & X) jeweils in Pfeilrichtung.

SPINNEN UM ZU ÜBERLEBEN

Einige Sekunden nach dem Programmstart erscheint auf dem Bildschirm in Großbuchstaben die Schrift „SPIDER GAME", neben der sich eine Spinne an ihrem Faden über den Fernseher herabläßt. Für die Titelmelodie mußte Beethoven mit einem Teil seiner Neunten herhalten. Ist die Melodie zu Ende, wird der Spieler gefragt, ob er eine Spielanleitung benötigt. Drückt man nun die J-Taste, so bekommt man eine dreiseitige Beschreibung von SPIDER GAME vorgesetzt, in der durch Betätigen einer Taste umgeblättert wird. Benötigt man keine Anleitung und teilt das dem Computer über die N-Taste mit, erscheint nach kurzer Wartezeit ein grünes Spielfeld. Wie auch immer man sich entscheidet, in jedem Fall wird bei der Tastaturabfrage kontrolliert, ob die Alpha-Lock-Taste noch nicht gelöst ist, und gegebenenfalls darauf hingewiesen. Geübte SPIDER-GAME-Spieler, die das Angebot einer Spielanleitung als Unterschätzung ihrer Fähigkeiten auffassen, können, indem sie den Joystick gleich nach dem Programmstart nach oben drücken, das Titelbild umgehen. Das funktioniert absichtlich nur bei gelöstem Alphalock.

Ist man nun bis zum Spielfeld vorgedrungen, so kann man an dessen Oberkante Energie und Lebenszeit einer in dessen Mitte befindlichen Spinne ablesen. Von nun an werden die anfangs 300 Energieeinheiten immer weniger, und die Lebenszeit nimmt in gleichem Maße zu. Wenn die Energie zu Ende ist, ist auch das Spiel beendet, und die Lebenszeit wird in Punkten gutgeschrieben. Man kann seinen Energievorrat jedoch immer wieder auffrischen, indem man Fliegen fängt, was Spinnen ja bekanntlich mit Netzen tun. So muß auch unsere Videospinne mit dem Faden, den sie beim Fortbewegen hinter sich herspinnt, ein möglichst dichtes Netz weben. Bewegt sich eine Fliege über das Netz, muß der Videospieler seine Spinne ruhig halten, da das Opfer sonst nicht in die Falle geht. Nach einem erfolgreichen Beutezug muß man nur noch zur Fliege hinkrabbeln und sie fressen, was man durch kurzes Anhalten über ihr erreicht. Das bringt dann 50 Einheiten Energie. Leider machen Ihnen beim Kampf ums Überleben auch noch kleine Kinder das Leben schwer, die mit Steinen auf Ihr kunstvolles Netz werfen. Die Steine reißen Löcher und befreien bereits gefangene Fliegen. Wird die Spinne von einem solchen Geschoß getroffen, ist das Spiel beendet. Das ist übrigens auch der Fall, wenn sie das Spielfeld verläßt.

Am Spiel ende bekommt man per Bildschirm mitgeteilt, auf welche Art die Spinne umgekommen ist. Man wird über den aktuellen Highscore informiert und gefragt, ob man noch einmal spielen möchte

PROGRAMMTECHNISCHES

In dem Programmteil TITELBILD findet man eine Routine, mit der sich bis zu 8 vergrößerte Buchstaben nebeneinander auf dem Bildschirm plazieren lassen. Normalerweise sind nur 4 Stück möglich, da nicht mehr Sprites in einer Reihe stehen dürfen. Im Hauptprogramm wird im Prinzip nur kontrolliert, ob die Spinne eine neue Schreibstelle erreicht hat, und abhängig von dem darauf befindlichen Fadensymbol, der Joystickeingabe sowie der ursprünglichen Bewegungsrichtung das neue Fadensymbol berechnet und ausgedruckt. Um langwierige IF-THEN-Abfragen zu vermeiden, sind die bei den verschiedenen Kombinationen auftretenden Symbole in einem Stringfeld (Z$) abgespeichert. Auch das Feststellen von Spielfeldübertretungen mittels ON ERROR spart Zeit.

Außerdem ist im Hauptprogramm dafür gesorgt, daß bei unbewegter Spinne der Programmteil FLIEGE FANGEN angesprungen wird. Darüber hinaus wird in zufälligen Abständen die Bewegungsrichtung der Fliege geändert und werden Steine generiert. In Abständen von jeweils 50 Punkten erfolgt ein Sprung zur Anzeigeroutine.

FUNKTIONEN

250–420 Initialisierung 1
Prescan, Spinnen- und Fliegenchars
460–600 Titelbild
Titelschrift, Bewegung der Spinne, Melodie, Spielanleitungsabfrage
640–780 Initialisierung 2
Faden- und Steinchars, Farben, Wertzuweisungen, Spielfeldaufbau
820–1010 Heuptprogramm
Zählwerk, Joystick- und Positionsabfrage, Berechnung und Ausdruck des Fadensymbols, neue Spinnenparameter
1050–1130 Fliege fangen
Spinne stoppen, Fliege über dem Netz anhalten
1170–1210 Bewegung Fliege
Zufällige Zuweisung von Richtung und entsprechendem Pattern
1250–1280 Steuerung Stein
Generierung von Spirte mit zufälliger Geschwindigkeit vom unteren Bildrand
1320–1390 Netz zerreißen
Spinne getroffen ?, Faden löschen
1430–1460 Fliege fressen
Fliegensprite löschen, Punkte geben
1500–1540 Anzeige
Energie und Lebenszeit anzeigen, kontrollieren, ob Energie verbraucht ist
1580–1610 Fehlerroutine
Registriert Spielfeldübertritt
1650–1780 Spielende
Mitteilung von Abbruchursache und Leistung, Melodie, Neues Spiel?
1820–1890 Titelmelodie
Melodiedatas
1930–2070 Spielanleitung
Text, Umblättern
2110–2130 Joystick
Tastaturabfrage, Umsetzung

VARIABLENVERZEICHNIS

A$, B$, C$ . . . . . .Titelschrift
L1, L2 . . . . . .Schleifenparamter
TH, TD . . . . . .Titelmelodie
XS, YS, XP, YP . . . . . .Spinnenposition
P1, P2 . . . . . .Spinnenpattern
XJ1, YJ1, XJ, YJ . . . . .Bewegungsdaten Spinne
K, S . . . . . .Tastaturdaten
FP, FX, FY . . . . . .Fligenparameter
Z, Z1, Z$, F . . . . . .Bestimmung von Fadensymbol
C1, C2 . . . . . .Coincidence
LZ . . . . . .Lebenszeit
HS . . . . . .Highscore
EN . . . . . .Energie
R . . . . . .Zufallszahl
D . . . . . .Entfernung
M . . . . . .Spinne bewegt sich
EF1, EF2 . . . . . .Fliege existiert
ES . . . . . .Sein existiert
ER . . . . . .Fehler ist aufgetreten
EC, ET, S, LN . . . . . .Fehlerdaten

Bitte lesen Sie weiter auf Seite 49

```
100 ! ************************
110 ! *   SPINNEN  SPIEL     *
120 ! * COPYRIGHT 1985  BY  *
130 ! *  Wolfgang Furtner   *
140 ! *  .bearbeitet vom    *
150 ! *       TEAM des      *
160 ! *  AKTUELL VERLAGES   *
170 ! *       MUENCHEN      *
180 ! * BENOETIGTE GERAETE  *
190 ! * TI 99/4A - KONSOLE  *
200 ! * TI  EXTENDED BASIC  *
210 ! * KASSETTEN-RECORDER  *
220 ! *      & JOYSTICK     *
230 ! *                     *
240 ! *    BENOETIGTER      *
250 ! *   SPEICHERPLATZ     *
260 ! *     9417 BYTES      *
270 ! *                     *
280 ! ************************
290 ! INITIALISIERUNG 1
300 GOTO 340 :: A$,B$=C$ ::
L1,L2,TH,TD,YP,XP,K,S,XJ,YJ,
XS,YS,C2,M,Z,F,R,D,EC,ET,LN=
HS
310 CALL CLEAR :: CALL COLOR
 :: CALL GCHAR :: CALL HCHAR
 :: CALL JOYST :: CALL KEY :
: CALL SCREEN
320 CALL CHAR :: CALL COINC
:: CALL DELSPRITE :: CALL MA
GNIFY :: CALL LOCATE :: CALL
 MOTION
330 CALL PATTERN :: CALL POS
ITION :: CALL SPRITE :: CALL
 CHARPAT :: CALL CHARSET ::
CALL ERR :: CALL SOUND
340 DIM Z$(5),FP(3),FX(3),FY
(3),Z1(1,1):: !$P-
350 CALL CLEAR :: RANDOMIZE
:: CALL SCREEN(2):: CALL MAG
NIFY(4)
360 CALL CHAR(96,"0000000006
0F0F07040202010000000000000
0000806010907040C")!FLIEGE O
BEN LINKS
370 CALL CHAR(100,"000000000
001060809 0E0203000000000000
00060F0F0E02040408")!FLIEGE
OBEN RECHTS
380 CALL CHAR(104,"000000000
1020204070F0F060000000000000
000C040709010608")!FLIEGE UN
TEN LINKS
390 CALL CHAR(108,"000000000
3020E0908060100000000000000
00080404020E0F0F06")!FLIEGE
UNTEN RECHTS
400 CALL CHAR(112,"012313896
71F033F4993172727470301 80C4C
891E6F8C0FC92C9E8E4E4E2C080"
)!SPINNE OBEN
410 CALL CHAR(116,"21130B098
77F037F8913171717270301 84C8D
090E1FEC0FE91C8E8E8E8E4C080"
)
420 CALL CHAR(120,"010347272
71793493F031F678913230180C0E
2E4E4E8C992FCC0F8E691C8C480"
)!SPINNE UNTEN
430 CALL CHAR(124,"010327171
71713897F037F87090B132180C0E
4E8E8E8C891FEC0FEE190D0C884"
)
440 CALL CHAR(128,"100849251
50D6FFFFF6F0D15254908104084 1
860803C7EFFFF7E3C8060188440"
)!SPINNE LINKS
450 CALL CHAR(132,"080585453
50D6FFFFF6F0D354585050880000
478803C7EFFFF7E3C8078040080"
)
460 CALL CHAR(136,"022118060
13C7EFFFF7E3C010618210208109
2A4A8B0F6FFFFF6B0A8A4921008"
)!SPINNE RECHTS
470 CALL CHAR(140,"0100201E0
13C7EFFFF7E3C011E20000110A0A
1A2ACB0F6FFFFF6B0ACA2A1A010"
)
480 ! TITELBILD
490 CALL JOYST(1,XJ,YJ):: IF
 YJ=4 THEN 650
500 A$="S PGIADMEER "
510 FOR L1=0 TO 2 :: C$="" :
: FOR L2=1 TO 4
520 CALL CHARPAT(ASC(SEG$(A$
,4*L1+L2,1)),B$):: C$=C$&B$
530 NEXT L2 :: CALL CHAR(36+
L1*4,C$):: NEXT L1
540 CALL SPRITE(#1,36,13,70,
64,#2,40,13,70,96,#3,44,13,7
0,128)
550 CALL CHAR(33,RPT$("08",8
)):: CALL COLOR(1,16,2):: CA
LL SPRITE(#4,120,7,1,190,2,0
):: CALL HCHAR(1,26,33)
560 FOR L1=1 TO 62 :: READ T
H,TD :: CALL SOUND(TD*1.6,TH
,4,TH*2,4,TH*3,4)
570 IF TD=100 THEN 590
580 CALL POSITION(#4,YP,XP):
: IF YP>8*Z THEN Z=Z+1 :: CA
LL HCHAR(Z+1,26,33)
```

```
590 NEXT L1
600 CALL MOTION(#4,0,0):: FO
R L1=2 TO 8 :: CALL COLOR(L1
,11,2):: NEXT L1
610 DISPLAY AT(22,1):"WILLST
 DU EINE":"SPIELANLEITUNG?":
"DRUECKE >J< ODER >N<"
620 CALL KEY(0,K,S):: IF K<9
6 AND K>65 THEN DISPLAY AT(2
1,1):"ALPHALOCK LOESEN" :: C
ALL SOUND(100,110,0)
630 IF K=106 THEN GOSUB 1720
 ELSE IF K<>110 THEN 620
640 ! INITIALISIERUNG 2
650 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL MAGNIFY(3)
660 CALL CHAR(33,RPT$("08",8
),34,"000000FF",35,"080808FF
08080808",36,"000000F8080808
08",37,"0000000F08080808",38
,"080808F8")
670 CALL CHAR(39,"0808080F",
40,"080808F808080808",41,"08
08080F08080808",42,"080808FF
",43,"000000FF08080808")
680 CALL CHAR(44,"0000000E1F
3F7F7F7F7F3F1F07000000000000
0038FCFEFEFEFCF8F0E")!STEIN
690 CALL COLOR(1,7,4,2,7,4):
: FOR L1=3 TO 8 :: CALL COLO
R(L1,3,12):: NEXT L1
700 FP(0)=96 :: FP(1)=100 ::
 FP(2)=104 :: FP(3)=108 :: F
X(1),FX(3),FY(2),FY(3)=12 ::
 FX(0),FX(2),FY(0),FY(1)=-12
710 Z1(0,0)=2 :: Z1(1,0)=5 :
: Z1(0,1)=3 :: Z1(1,1)=4
720 Z$(0)="!!##()()()##" ::
Z$(1)=CHR$(34)&"#"&CHR$(34)&
"#++**###*+" :: Z$(2)="&(*#(#
&*(#*#"
730 Z$(3)="')*##)*'#)*#" ::
Z$(4)="%)+#+%#)#)#+" :: Z$(5
)="$(+#$+(#(##+"
740 !$P+
750 EN=300 :: P1=112 :: P2=1
16 :: ER,C1,LZ,EF1,EF2,ES,XS
1,YS1,XJ1=0 :: YJ1=4
760 !$P-
770 ON ERROR 1430
780 CALL SPRITE(#5,112,2,93,
125)
790 DISPLAY AT(1,1):"ENERGIE
:" :: DISPLAY AT(1,16):"ZEIT
:"
800 ! HAUPTPROGRAMM
810 IF INT(EN/50)=EN/50 THEN
 GOSUB 1370
820 IF C1 THEN 1480
830 EN=EN-1 :: LZ=LZ+1
840 CALL JOYST(1,XJ,YJ):: IF
 XJ=0 AND YJ=0 THEN M=0 :: G
OTO 1020
850 CALL PATTERN(#5,P1)
860 IF XJ THEN YJ=0
870 M=-1 :: CALL MOTION(#5,-
YJ1,XJ1)
880 CALL PATTERN(#5,P2)
890 CALL POSITION(#5,YP,XP):
: XS=INT((XP+7)/8)+1 :: YS=I
NT((YP+7)/8)+1
900 IF XS=XS1 AND YS=YS1 THE
N GOSUB 1120 :: GOTO 890
910 IF XJ1 THEN IF XJ THEN Z
=1 :: GOTO 930 ELSE Z=Z1(-(Y
J=-4),-(XJ1=-4)):: GOTO 930
920 IF YJ THEN Z=0 ELSE Z=Z1
(-(YJ1=4),-(XJ=4))
930 CALL PATTERN(#5,P1)
940 CALL GCHAR(YS,XS,F)
950 CALL HCHAR(YS,XS,ASC(SEG
$(Z$(Z),F-31,1)))
960 CALL PATTERN(#5,P2)
970 P1=112-8*(YJ=-4)-16*(XJ=
-4)-24*(XJ=4):: P2=P1+4
980 IF Z>1 THEN CALL MOTION(
#5,0,0):: CALL LOCATE(#5,YS*
8-12,XS*8-11)
990 XJ1=XJ :: YJ1=YJ
1000 XS1=XS :: YS1=YS :: GOT
O 810
1010 ! FLIEGE FANGEN
1020 CALL MOTION(#5,0,0)
1030 IF RND<.3 THEN GOSUB 11
20
1040 CALL COINC(#5,#8,10,C2)
:: IF C2 THEN 1320
1050 IF EF2 THEN 810
1060 CALL POSITION(#3,YP,XP)
1070 YS=INT((YP+7)/8)+1 :: X
S=INT((XP+7)/8)+1
1080 IF XS>2 AND XS<30 AND Y
S>1 AND YS<24 THEN CALL GCHA
R(YS,XS,F)ELSE 810
1090 IF F=35 THEN CALL DELSP
RITE(#3):: CALL SPRITE(#8,FP
(R),2,YP,XP):: EF1=0 :: EF2=
-1
1100 GOTO 810
1110 ! BEWEGUNG FLIEGE
```

```
1120 IF RND<.2 THEN 1180
1130 IF ES AND RND<.2 THEN 1
230
1140 R=INT(RND*4)
1150 IF EF1 THEN CALL MOTION
(#3,FY(R),FX(R)):: CALL PATT
ERN(#3,FP(R))ELSE CALL SPRIT
E(#3,FP(R),2,220,100,FY(R),F
X(R)):: EF1=-1
1160 RETURN
1170 ! STEUERUNG STEIN
1180 CALL POSITION(#5,YP,XP)
1190 XP=XP+INT(RND*2*D)-D ::
 IF XP<13 THEN XP=13 ELSE IF
 XP>235 THEN XP=235
1200 CALL SPRITE(#4,44,15,20
0,XP,-INT(RND*20)-10,0):: CA
LL SOUND(-50,110,0,-5,0)
1210 ES=-1 :: RETURN
1220 ! NETZ ZERREISSEN
1230 CALL POSITION(#4,YP,XP)
:: IF YP<25 OR YP>175 THEN R
ETURN
1240 CALL MOTION(#5,0,0):: C
ALL SOUND(-200,-5,0,110,0)
1250 CALL COINC(#4,#8,25,C2)
:: IF C2 THEN CALL DELSPRITE
(#8):: EF2=0
1260 CALL COINC(#4,#5,12,C1)
:: IF C1 THEN CALL MOTION(#4
,0,0):: RETURN
1270 XS=INT((XP+3)/8)+1 :: Y
S=INT((YP+3)/8)+1
1280 CALL HCHAR(YS-2,XS-1,32
,2):: CALL HCHAR(YS-1,XS-2,3
2,4):: CALL HCHAR(YS,XS-2,32
,4):: CALL HCHAR(YS+1,XS-1,3
2,2)
1290 ES=0 :: CALL DELSPRITE(
#4):: IF M THEN CALL MOTION(
#5,-YJ1,XJ1)
1300 RETURN
1310 ! FLIEGE FRESSEN
1320 CALL SOUND(-400,-4,0,55
5,5)
1330 CALL DELSPRITE(#8):: EF
2=0
1340 EN=EN+50 :: GOSUB 1370
1350 GOTO 810
1360 ! ANZEIGE
1370 DISPLAY AT(1,21)SIZE(5)
:USING "######":LZ
1380 DISPLAY AT(1,9)SIZE(5):
USING "######":EN
1390 IF EN<=0 THEN 1480
1400 D=50-LZ/50 :: IF D<10 T
HEN D=10
1410 RETURN
1420 ! FEHLERROUTINE
1430 ER=-1
1440 CALL ERR(EC,ET,S,LN)
1450 IF EC=79 AND LN=950 THE
N 1480
1460 CALL CLEAR :: PRINT "FE
HLER";EC;"IN ZEILE";LN :: EN
D
1470 ! SPIELENDE
1480 FOR L1=888 TO 111 STEP
-111 :: CALL SOUND(500-L1/2,
L1,0):: NEXT L1
1490 CALL DELSPRITE(ALL):: C
ALL CLEAR
1500 IF C1 THEN DISPLAY AT(3
,1):"DU WURDEST VON EINEM" :
: DISPLAY AT(5,1):"STEIN ERS
CHLAGEN" :: GOTO 1530
1510 IF ER THEN DISPLAY AT(3
,1):"DU HAST DAS SPIELFELD"
:: DISPLAY AT(5,1):"VERLASSE
N" :: GOTO 1530
1520 DISPLAY AT(3,1):"DEINE
ENERGIE IST" :: DISPLAY AT(5
,1):"ERSCHOEPFT"
1530 DISPLAY AT(8,1):USING "
PUNKTE:....#####":LZ :: DISP
LAY AT(10,1):USING "HIGHSCOR
E: #####":HS
1540 IF HS<LZ AND HS<>0 THEN
 DISPLAY AT(13,1):"DU HAST E
INEN NEUEN" :: DISPLAY AT(15
,1):"REKORD AUFGESTELLT"
1550 HS=MAX(HS,LZ)
1560 DISPLAY AT(19,1):"NOCH
EIM SPIEL?" :: DISPLAY AT(21
,1):"DRUECKE >J< ODER >N<"
1570 RESTORE 1640 :: FOR L1=
1 TO 62 :: CALL KEY(0,K,S)
1580 IF K=106 OR K=74 THEN C
ALL CLEAR :: GOTO 750
1590 IF K=110 OR K=78 THEN E
ND
1600 READ TH,TD :: CALL SOUN
D(TD*1.6,TH,4,TH*2,4,TH*3,4)
1610 NEXT L1 :: GOTO 1570
1620 ! TITELMELODIE
1630 !§P+
1640 DATA 330,200,330,200,34
9,200,392,200,392,200,349,20
0,330,200,294,200,262,200,26
2,200,294,200,330,200
```

```
1650 DATA 330,300,294,100,29
4,300,330,200,330,200,349,20
0,392,200,392,200,349,200,33
0,200,294,200
1660 DATA 262,200,262,200,29
4,200,330,200,294,300,262,10
0,262,300,294,200,294,200,33
0,200,262,200
1670 DATA 294,200,330,100,34
9,100,330,200,262,200,294,20
0,330,100,349,100,330,200,29
4,200
1680 DATA 262,200,294,200,19
6,300,330,200,330,200,349,20
0,392,200,392,200,349,200
1690 DATA 330,200,294,200,26
2,200,262,200,294,200,330,20
0,294,300,262,100,262,300
1700 !§P-
1710 ! SPIELANLEITUNG
1720 CALL CLEAR :: CALL DELS
PRITE(ALL):: CALL CHARSET ::
 CALL MAGNIFY(3)
1730 FOR L1=1 TO 8 :: CALL C
OLOR(L1,11,2):: NEXT L1
1740 CALL SPRITE(#1,136,7,5,
48,#2,128,7,5,184):: DISPLAY
 AT(2,9):"SPIDER GAME"
1750 DISPLAY AT(5,1):"DU BIS
T EINE SPINNE, UND": :"WILLS
T MOEGLICHST LANGE": :"UEBER
LEBEN. DAZU HAST DU"
1760 DISPLAY AT(11,1):"300 E
NERGIEEINHEITEN ZUR": :"VERF
UEGUNG. DOCH DIE": :"SIND SC
HNELL VERBRAUCHT."
1770 DISPLAY AT(17,1):"UM DE
INE ENERGIEVORRAT AUF-": :"Z
UFRISCHEN MUSST DU": :"FLIEG
EN FANGEN." :: GOSUB 1840
1780 DISPLAY AT(5,1):"DIE FL
IEGEN WERDEN IN EINEM": :"NE
TZ GEFANGEN, DAS DU": :"BEIM
 FORTBEWEGEN HINTER"
1790 DISPLAY AT(11,1):"DIR H
ERSPINNST. DOCH NUR": :"DORT
 WO DAS NETZ DICHT IST": :"U
ND NUR WENN DU DICH NICHT"
1800 DISPLAY AT(17,1):"BEWEG
ST, GEHEN DIE FLIEGEN": :"IN
S NETZ. EINE FLIEGE BRINGT":
 :"50 ENERGIEEINHEITEN." ::
GOSUB 1840
1810 DISPLAY AT(5,1):"KLEINE
 KINDER DIE STEINE AUF": :"D
AS NETZ WERFEN ERSCHWEREN":
:"DIR DAS LEBEN. TRIFFT DICH
"
1820 DISPLAY AT(11,1):"EIN S
TEIN, VERLAESST DU DAS": :"S
PIELFELD ODER GEHT DEINE": :
"ENERGIE ZUENDE, SO IST DAS"
1830 DISPLAY AT(17,1):"SPIEL
 AUS. DIE SPINNE WIRD": :"MI
T DEM JOYSTICK GESTEUERT.":
:" *** VIEL GLUECK ! ***" ::
 GOSUB 1840 :: RETURN
1840 DISPLAY AT(23,1):"DRUEC
KE EINE TASTE!"
1850 CALL KEY(0,K,S):: IF S<
1 THEN 1850
1860 CALL HCHAR(5,1,32,640):
: RETURN
1870 ! JOYSTICK
1880 XJ,YJ=0 :: CALL KEY(0,K
,S)
1890 IF K=101 THEN YJ=4 ELSE
 IF K=120 THEN YJ=-4 ELSE IF
 K=115 THEN XJ=-4 ELSE IF K=
100 THEN XJ=4
1900 RETURN
```

Fortsetzung von Seite 45

SPRITES
Titelbild:
Nr. 1–3 . . . . . .Titelschrift
Nr. 4 . . . . . .Spinne
Spielanleitung:
Nr. 1, Nr. 2 . . . . . .Spinnen
Spielablauf:
Nr. 5 . . . . . .Spinne
Nr. 3 . . . . . .bewegliche Fliege
Nr. 8 . . . . . .gefangene Fliege
Nr. 4 . . . . . .Stein

CHARS
Titelbild:
33 . . . . . .Faden
36–47 . . . . . .Titelschrift
Spielablauf:
33–43 . . . . . .Faden
44–47 . . . . . .Stein
96–111 . . . . . .Fliege
112–143 . . . . . .Spinne

Wolfgang Furtner

```
100 ! ************************
110 ! *       HEINRICH       *
120 ! *         der          *
125 ! *Schreckensherrscher *
100
130 ! *    Copyright by      *
140 ! *Bernhard  Lobenhofer*
150 ! *                      *
160 ! *   Bearbeitet vom     *
170 ! *      Team des        *
180 ! *  Aktuell Verlages    *
190 ! *       Muenchen       *
200 ! *                      *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole    *
230 ! *     Ext. Basic       *
240 ! *    Joystick (1)      *
250 ! *                      *
270 ! *  Speicherbelegung    *
280 ! *     12596 Bytes      *
290 ! ************************
295 !
300 CALL CLEAR
310 CALL CHAR(91,"3C4299A1A1
99423C")
320 CALL SCREEN(4)
330 DISPLAY AT(10,1):".....
H E I N R I C H......."
340 DISPLAY AT(12,1):"...DER
 SCHRECKENSHERRSCHER.."
350 DISPLAY AT(22,1):" A NOV
. 1984 BY"
360 DISPLAY AT(23,1):" STEFA
N RIES  &"
370 DISPLAY AT(24,1):" BERNH
ARD LOBENHOFER"
380 FOR I=1 TO 750 :: NEXT I
390 CALL CLEAR
400 SCHW=0 :: CALL MAGNIFY(1
)
410 RANDOMIZE
420 CALL CLEAR
430 CALL SCREEN(8)
440 GOSUB 460
450 GOTO 830
460 CALL COLOR(1,12,1,2,16,1
,3,4,1,4,3,1,5,6,1,9,7,1,10,
7,1,11,7,1,13,15,1,12,15,1)
470 CALL COLOR(6,9,1,7,9,1)
480 CALL CLEAR
490 CALL CHAR(134,"0307070F1
F1F7F7F7F3F3F1F0F0F1F02E0F0F
8F8FCFCFEFEFEFCF8F8FCFC3C")
500 CALL CHAR(96,"FFFFFF49C9
49D24B",97,"01070F1F3F7F7FFF
",98,"0103071F3F3F7FFF",99,"
070F0F0F1F3F7FFF",100,"00000
00001030307")
510 CALL CHAR(101,"0F1F7FFFF
FFFFFFF",102,"00C0E0E0E0F0F0
F8",103,"F8FCFEFCFCFEFEFF",1
04,RPT$("F",16),105,"7FBFDFD
FE7FBFBFC")
520 CALL CHAR(106,"7FBFBFBFC
FF3FDFE",107,"FEF9FBE7DFBFBF
7F",108,"FEF1EFDFEFEF9F7F",1
09,"010303070F1F7FFF",110,"C
0C0E0E0E0F0FCFF")
530 CALL CHAR(111,"8080C0C0F
8FCFEFF",112,"C0F0F0F0F8FCFE
FF",35,"804020100B070F0F",36
,"FF0F070B10204080",37,"8182
8488D0E0F0F0")
540 CALL CHAR(38,"FFF0E0D088
848201",40,"1C3F7F7FFFFFFFFF
",41,"FFFFFFFF7F3F1F01",42,"
70FEFFFFFFFFFFFF",43,"FFFFFF
FFFFFFE3C1")
550 CALL CHAR(44,"3C7EFFFFFF
FFFEFE",45,"FEFEFFFFFFFFFE1C
",59,"183C1818FFBD1866",60,"
01030703070F1F07",61,"0F1F3F
7F03030303")
560 CALL CHAR(62,"80C0E0C0E0
F0F8E0",63,"F0F8FCFEC0C0C0C0
",64,"0000000000387CFF",48,"
0708081212222A16",49,"141412
0A0A0A0E03")
570 CALL CHAR(50,"C020101048
084424",51,"A4A8A8505090A0C0
",91,"3F7FFFCCCCFFFFFF",92,"
FFFFF3E1E1F1E1E1",93,"DBDBFF
FFE7E7E7E7")
580 CALL CHAR(94,RPT$("F",16
),113,"0000000000000207",114
,RPT$("F",16),65,RPT$("F",16
))
590 CALL CHAR(115,"030101050
3010103",116,"C0C0C0C0C0C0E0
C0")
600 CALL CHAR(120,"030404040
40303070509030700191909C020B
020A0C080C0E0A090808080C0"
,46,"191A040B13204080")
610 CALL CHAR(128,RPT$("F",1
6),66,RPT$("F",16))
620 CALL CHAR(131,"FFFF7F3F3
C180000",132,"FFFFFF3F3F0A00
00",133,"FFFF7F2E0E0C0400")
630 CALL CHAR(124,"030404040
40303070509010101010101C020B
020A0C0C0E0A090808080808C0"
)
```

```
640 CALL CHAR(136,"0307070F1
F1F7F7F7F3F3F1F0F0F1F02E0F0F
8F8FCFCFEFEFEFCF8F8FCFC3C00"
)
650 CALL CHAR(72,"0000000001
03060C",73,"18306060FFC0C0C0
",74,"C0C0C0C0C0C0C0C0",75,"
C0C0C0C0C0C0C0FF",76,"183C7E
DB99181818")
660 CALL CHAR(77,"18181818FF
181818",78,RPT$("18",8),79,"
1818181818181FF",80,"000000
0080C06030",81,"180C0606FF03
0303")
670 CALL CHAR(82,RPT$("03",8
),83,"0303030303030FF")
680 ! BERG ZEICHNEN
690 DISPLAY AT(12,1):"hhhhhh
hhhhhhhhhhhhhhhhhhhh"
700 DISPLAY AT(11,1):"hhhhhh
hhhhhhhhhhhhhhhhhhhh"
710 DISPLAY AT(10,1):"ahhihh
hhhhhhhhhkhhhhhhhhjhhh"
720 DISPLAY AT(9,1):" mnbhhh
hhhhhhhhhlhhhhhhihhhp"
730 DISPLAY AT(8,1):"....ahh
hhhhhhhhhobhhhhpbhho."
740 DISPLAY AT(7,1):".....bh
hhhhjhhho..ahho..mn.."
750 DISPLAY AT(6,1):"......b
hhhjhhhp....mn......."
760 DISPLAY AT(5,1):".......
chgbhhp............."
770 DISPLAY AT(4,1):".......
def.mn..............."
780 DISPLAY AT(3,1):".......
.q..................."
790 ! WOLKEN & SONNE MALEN
800 DISPLAY AT(1,1):"....(*,
.......(*,.........#%"
810 DISPLAY AT(2,1):"....)+-
.......)+-.........$&"
820 RETURN
830 DISPLAY AT(14,27):" Ü" :
: DISPLAY AT(15,27):"Ä^" ::
DISPLAY AT(16,27):"ö^"
840 CALL HCHAR(22,1,64,32)::
 CALL HCHAR(23,1,65,64)
850 CALL HCHAR(17,25,48):: C
ALL HCHAR(18,25,49):: CALL H
CHAR(17,26,50):: CALL HCHAR(
18,26,51):: CALL HCHAR(19,25
,115):: CALL HCHAR(19,26,116
)
860 CALL SPRITE(#1,59,14,122
,20,0,4):: GOSUB 1940
870 CALL COINC(#1,122,235,15
,C0):: IF C0=-1 THEN 880 ELS
E 870
880 !
890 CALL DELSPRITE(#1)
900 FOR I=1 TO 250 :: NEXT I
 :: CALL CLEAR :: CALL SCREE
N(2)
910 CH=130
920 CALL HCHAR(1,1,128,160)
930 SP=1
940 ZU=INT(RND*(133-131+1))+
131 :: CALL HCHAR(6,SP,ZU,1)
:: SP=SP+1 :: IF SP<>33 THEN
 940 ELSE 950
950 DISPLAY AT(11,1):"...HLP
......HLP......HLP.......IMQ
......IMQ......IMQ.......JNR
......JNR......JNR.......KOS
......KOS......KOS...."
960 CALL HCHAR(24,1,66,32)
970 CALL MAGNIFY(3)
980 CALL SPRITE(#2,136,15,48
,20,30,0,#3,136,15,48,60,38,
0)
990 CALL SPRITE(#4,136,15,48
,85,35,0,#5,136,15,48,109,50
,0,#6,136,15,48,135,20,0)
1000 CALL SPRITE(#7,136,15,4
8,155,40,0,#8,136,15,48,180,
35,0,#9,136,15,48,210,35,0)
1010 CALL SPRITE(#10,136,15,
48,230,50,0)
1020 CALL HCHAR(23,17,46)
1030 CALL SPRITE(#1,120,3,16
9,2,0,0)
1040 CALL JOYST(1,X,Y)
1050 CALL PATTERN(#1,124)
1060 CALL MOTION(#1,0,X*4)
1070 CALL SOUND(-10,440,5,88
0,5,-4,0)
1080 CALL PATTERN(#1,120)
1090 CALL COINC(#1,184,136,2
5,C1):: IF C1=-1 THEN 1100 E
LSE 1120
1100 SCHW=1 :: CALL SOUND(10
0,1047,0,110,0,-5,0)
1110 CALL HCHAR(23,17,32)
1120 CALL COINC(ALL,C):: IF
C=-1 THEN 1140 ELSE 1130
1130 CALL COINC(#1,182,256,2
5,C2):: IF (C2=-1)AND(SCHW=1
)THEN 1200 ELSE IF (C2=-1)AN
D(SCHW=0)THEN 1030 ELSE 1040
1140 ! ENDE
```

```
1150 CALL DELSPRITE(ALL):: C
ALL CLEAR :: CALL CHARSET ::
 CALL SCREEN(12)
1160 DISPLAY AT(4,1):".....
.G A M E..O V E R....."
1170 DISPLAY AT(12,1):"DURCH
 IHR TRAGISCHES ENDE...KONNT
EN SIE LEIDER NICHT....VERHI
NDERN,DEN GRAUSAMEN....HERRS
CHER VOM THRON ZU....."
1180 DISPLAY AT(16,1):"STUER
ZEN." :: DISPLAY AT(19,1):".
.....S C H A D E !!!....."
1190 GOSUB 1940 :: GOSUB 216
0 :: CALL CLEAR :: END
1200 CALL CLEAR :: CALL SCRE
EN(8):: CALL DELSPRITE(ALL):
: GOSUB 460
1210 CALL CHAR(48,"4850E0444
2C14261",49,"70188C6C66723B0
F",50,"412316241C183061",51,
"E16366CCDCD8DCF0")
1220 CALL CHAR(52,"080606633
160C061",53,"331A2CC5050D1F0
F",54,"244891128CC8C891",55,
"1624EC0B1820F0F0")
1230 CALL CHAR(88,"000000010
7070F0F",89,"1F1F0F0F1F7F7FF
F",90,"073F7FFFFFFFFFFF",91,
"F8F0F0F0F0F0F0F0",92,"80C0F
EFFFFFFFFFF")
1240 CALL CHAR(93,"1F1F1F1F1
F1F1F1F",94,"000000C0E0F0F8F
8",95,"FCFCFCFEFEFFFFFF")
1250 CALL CHAR(72,"000010207
0B1F32F1F0B09143600000000000
03C78F1E1FEFCC0E0F0783C0000"
)
1260 CALL CHAR(77,"000042FF4
242FF42")
1270 CALL HCHAR(14,12,52)::
CALL HCHAR(15,12,53):: CALL
HCHAR(14,13,54):: CALL HCHAR
(15,13,55)
1280 CALL HCHAR(15,20,48)::
CALL HCHAR(16,20,49):: CALL
HCHAR(15,21,50):: CALL HCHAR
(16,21,51)
1290 CALL HCHAR(19,25,52)::
CALL HCHAR(20,25,53):: CALL
HCHAR(19,26,54):: CALL HCHAR
(20,26,55)
1300 CALL HCHAR(13,1,77,32):
: CALL HCHAR(24,1,77,32):: C
ALL VCHAR(14,1,77,10):: CALL
 VCHAR(14,32,77,10)
1310 CALL SPRITE(#3,72,13,98
,256,0,-30,#4,72,2,120,256,0
,-25)
1320 CALL SPRITE(#5,72,13,14
0,256,0,-35,#6,72,2,160,256,
0,-19,#7,72,5,174,256,0,-40)
1330 CALL SPRITE(#1,130,5,10
5,9,0,0)
1340 BEW=0
1350 CALL JOYST(1,X,Y)
1360 IF BEW=75 THEN 1430
1370 CALL MOTION(#1,-Y*5,0)
1380 CALL SOUND(-10,349,5,65
9,5,-8,0)
1390 CALL PATTERN(#1,120)
1400 CALL POSITION(#1,Y,X)::
 IF Y>180 OR Y<104 THEN 2070
1410 BEW=BEW+1 :: CALL COINC
(ALL,CA):: IF CA=-1 THEN 207
0 ELSE 1420
1420 GOTO 1350
1430 CALL CLEAR :: CALL DELS
PRITE(ALL):: CALL SCREEN(2)
1440 CALL CHARSET
1450 CALL CHAR(128,"0A070204
04040301011F23270D191905A8F0
209018D420C8C8F880808080C0
")
1460 CALL CHAR(40,"030707070
70304C8",41,"FF2422100804030
0",42,"C0E0E0E0E0C02013",43,
"FF1414181030C000")
1470 CALL CHAR(48,"050585432
191492191492195452513O1A0A0A
1C284899284899284A9A2A4C880"
)
1480 CALL CHAR(124,"03040404
040303070509030700191909C020
B020A0C080C0E0A090808080C0
")
1490 CALL CHAR(53,"0000E0F0F
8FCFEFF",54,RPT$("F",16),55,
"0000070F1F3F7FFF",56,"00000
70810204080",57,RPT$("80",8)
)
1500 CALL CHAR(58,"808183878
F9FBFFF",59,"0000E0100804020
1",60,"01010307OF1F3F7F",61,
RPT$("F",16))
1510 CALL CHAR(136,"0000070F
1F3F7FFF",137,RPT$("F",16),1
38,"0000E0F0F8FCFEFF")
1520 CALL COLOR(2,16,1,4,15,
1,3,14,1,14,14,1)
```

```
1530 CALL HCHAR(22,29,53)::
CALL HCHAR(23,29,54):: CALL
HCHAR(24,29,54):: CALL HCHAR
(22,30,56):: CALL HCHAR(23,3
0,57):: CALL HCHAR(24,30,58)
1540 CALL HCHAR(22,31,59)::
CALL HCHAR(23,31,60):: CALL
HCHAR(24,31,61):: CALL HCHAR
(22,32,55):: CALL HCHAR(23,3
2,54):: CALL HCHAR(24,32,54)
1550 CALL HCHAR(1,4,40):: CA
LL HCHAR(2,4,41):: CALL HCHA
R(1,5,42):: CALL HCHAR(2,5,4
3)
1560 CALL HCHAR(1,8,40):: CA
LL HCHAR(2,8,41):: CALL HCHA
R(1,9,42):: CALL HCHAR(2,9,4
3)
1570 CALL HCHAR(1,12,40):: C
ALL HCHAR(2,12,41):: CALL HC
HAR(1,13,42):: CALL HCHAR(2,
13,43)
1580 CALL HCHAR(1,16,40):: C
ALL HCHAR(2,16,41):: CALL HC
HAR(1,17,42):: CALL HCHAR(2,
17,43)
1590 CALL SPRITE(#2,48,16,8,
27,55,0,#3,48,16,8,59,35,0,#
4,48,16,8,91,40,0)
1600 CALL SPRITE(#5,48,16,8,
123,50,0)
1610 CALL SPRITE(#6,128,11,1
77,135,0,4)
1620 CALL SPRITE(#1,124,3,17
7,3,0,0)
1630 CALL JOYST(1,X,Y)
1640 CALL MOTION(#1,0,X*4)
1650 CALL POSITION(#1,Y,X)::
 IF X<5 THEN 1620 ELSE 1660
1660 CALL COINC(#1,#2,15,S1)
:: IF S1=-1 THEN 2120 ELSE C
ALL COINC(#1,#3,15,S2):: IF
S2=-1 THEN 2120 ELSE CALL CO
INC(#1,#4,15,S3):: IF S3=-1
THEN 2120 ELSE 1670
1670 CALL COINC(#1,#5,19,S4)
:: IF S4=-1 THEN 2120 ELSE C
ALL COINC(#1,#6,10,S5):: IF
S5=-1 THEN 1690 ELSE CALL CO
INC(#6,177,237,8,S6):: IF S6
=-1 THEN 1830 ELSE 1680
1680 GOTO 1630
1690 CALL DELSPRITE(ALL):: C
ALL CLEAR :: FOR I=1 TO 16 :
: CALL SCREEN(I):: NEXT I
1700 DISPLAY AT(4,1):".....
S U P E R !!!!.............
--------------........."
1710 DISPLAY AT(12,1):"SIE H
ABEN DEN FURCHTBARSTEN HERRS
CHER ALLER ZEITEN BE-..SIEGT
 UND SOMIT DAS VOLK....BEFRE
IT."
1720 FOR I=1 TO 2
1730 CALL SOUND(500,131,0,16
5,0,196,0):: CALL SOUND(500,
131,0,175,0,220,0)
1740 CALL SOUND(500,131,0,16
5,0,196,0):: CALL SOUND(500,
123,0,147,0,196,0):: CALL SO
UND(500,131,0,165,0,196,0)
1750 CALL SOUND(500,165,0,19
6,0,262,0):: CALL SOUND(500,
175,0,220,0,262,0):: CALL SO
UND(500,165,0,196,0,262,0)
1760 CALL SOUND(500,147,0,19
6,0,247,0)
1770 CALL SOUND(500,165,0,19
6,0,262,0):: CALL SOUND(500,
196,0,262,0,330,0):: CALL SO
UND(500,220,0,262,0,349,0)
1780 CALL SOUND(500,196,0,26
2,0,330,0)
1790 CALL SOUND(500,196,0,24
7,0,294,0):: CALL SOUND(500,
196,0,262,0,330,0)
1800 NEXT I
1810 GOSUB 2160
1820 CALL CLEAR :: END
1830 CALL DELSPRITE(ALL)
1840 CALL VCHAR(22,29,32,3):
: CALL VCHAR(22,32,32,3):: C
ALL HCHAR(22,30,136):: CALL
HCHAR(23,30,137):: CALL HCHA
R(24,30,137)
1850 CALL HCHAR(22,31,138)::
 CALL VCHAR(23,31,137,2)
1860 FOR I=1 TO 150 :: NEXT
I :: CALL CLEAR :: CALL SCRE
EN(11):: CALL CHARSET
1870 DISPLAY AT(4,1):".....
.G A M E..O V E R....."
1880 DISPLAY AT(12,1):"KURZ
BEVOR SIE DEN KOENIG...ERWIS
CHEN KONNTEN, ENTKAM...ER DU
RCH EINEN GEHEIMGANG. ."
1890 DISPLAY AT(17,1):"SO WI
RD ER WEITER SEINE.....SCHRE
CKENSHERRSCHAFT VER-...BREIT
EN........."
```

```
1900 FOR I=1 TO 3
1910 CALL SOUND(1000,131,0,1
23,0,110,0):: CALL SOUND(100
0,147,0,165,0,175,0):: CALL
SOUND(1000,110,0,123,0,131,0
)
1920 NEXT I
1930 GOSUB 2160 :: CALL CLEA
R :: END
1940 RESTORE 1990
1950 FOR I=1 TO 50
1960 READ DA,FR1,FR2
1970 CALL SOUND(DA,FR1,0,FR2
,0)
1980 NEXT I
1990 DATA 250,185,30000,125,
165,30000,250,147,30000,125,
139,30000,250,147,30000,125,
139,30000,250,147,30000
2000 DATA 125,185,30000,250,
247,30000,125,220,30000,750,
139,165,250,139,165,125,147,
185,250,165,196
2010 DATA 125,156,185,250,16
5,196,125,220,185,250,165,19
6,125,185,220,250,220,277,12
5,196,247,750,147,185
2020 DATA 250,185,30000,125,
220,30000,250,294,30000,125,
277,30000,375,185,220,375,18
5,294,250,220,277
2030 DATA 125,196,247,250,24
7,294,125,247,294,750,196,27
7,250,196,247,125,185,220,25
0,220,277
2040 DATA 125,196,247,375,19
6,247,375,196,220,250,139,16
5,125,139,165,250,165,196,12
5,185,30000,750,147,185
2050 DATA 250,139,165,125,13
9,165,250,123,196,125,139,18
5,375,147,30000,125,185,220
2060 RETURN
2070 CALL DELSPRITE(ALL):: C
ALL CHARSET :: CALL CLEAR ::
 CALL SCREEN(11)
2080 DISPLAY AT(4,1):".....
G A M E..O V E R....."
2090 DISPLAY AT(12,1):"SIE W
ARENIHREM ZIEL SCHON..RECHT
 NAHE.DOCH DER SCHRECK-LICHE
 TYRANN SANDTE SEINE...SCHRE
CKENSBOTEN."
2100 DISPLAY AT(18,1):"UND D
IESE KENNEN KEIN.....ERBAR
MEN..... "
2110 GOSUB 1940 :: GOSUB 216
0 :: CALL CLEAR :: END
2120 CALL DELSPRITE(ALL):: C
ALL CHARSET :: CALL CLEAR ::
 CALL SCREEN(11)
2130 DISPLAY AT(4,1):".....
.G A M E..O V E R....."
2140 DISPLAY AT(12,1):"FAST
HAETTEN SIE IHR ZIEL...ERREI
CHT! DOCH DIE BOGEN-...SCHUE
TZEN DES TYRANNS.....WAREN
 SCHNELLER....."
2150 GOSUB 1940 :: GOSUB 216
0 :: CALL CLEAR :: END
2160 DISPLAY AT(23,1):"NEUES
 SPIEL ?.............(FEUE
RKNOPF DRUECKEN!)"
2170 CALL KEY(1,K,S):: IF S=
0 THEN 2170 :: IF K=18 THEN
400
2180 RETURN
```

HEINRICH DER SCHRECKENSHERRSCHER

Du hast die ehrenvolle Aufgage übernommen, den schrecklichen Tyrannen Heinrich von seinem Thron zu stürzen.
Ohne Schwierigkeiten erreichst Du die Burg des Schrecklichen. Doch jetzt heißt es aufgepaßt. Überall lauern Gefahren auf Dich. Nachdem Du die Zugbrücke überschritten hast, brechen die Festungsmauern über Dir zusammen. Wenn es Dir gelingt, mit dem Schwert in der Mitte des Burghofes die andere Seite zu erreichen, ohne von den herabfallenden Trümmern getroffen zu werden, gelangst Du in den Garten der Drachen. Ihnen mußt Du solange widerstehen, bis sich das Tor zum Schloß für Dich öffnet. Berührst Du dabei den Zaun, ist Dein einziges Leben beendet. Jetzt wird der Herrscher nur noch von seinen Bogenschützen auf der Galerie verteidigt. In ihrem Schutz versucht er, den Eingang zu einem Geheimgang zu erreichen.
Um ihm endgültig den Garaus zu machen, mußt Du vor ihm an den Geheimgang gelangen. Doch die Bogenschützen schlafen nicht und Du mußt ganz schön aufpassen, um auch den letzten Teil dieses Spieles lebend zu überstehen und Deine Aufgabe zu erfüllen.

Bernhard Lobenhofer

VULKANIEN

Du bist mit Deinem Fesselballon durch mystische Winde in das geheimnisvolle Land Vulkanien abgetrieben worden. Durch giftige Wolken, Feuer und Lavasteine befindest Du Dich in großer Gefahr.
Sämtliche Vulkane sind ausgebrochen und die Einwohner verlassen in großer Eile ihr Land. Dir allein kann es gelingen, das Land zu retten. Fliege mit Deinem Ballon zweimal von links nach rechts über den Bildschirm, hüte Dich dabei aber vor den schwarzen Wolken und den ausgebrochenen Vulkanen. Hast Du diese ersten beiden Aufgaben gelöst, stehen Dir beim dritten Anflug 5 Wasserbomben zum Löschen der Vulkane zur Verfügung. In den

Bitte lesen Sie weiter auf Seite 60

```
100 ! ************************
110 ! *                      *
120 ! *     VULKANIEN        *
130 ! *    Copyright by      *
140 ! *  Werner  Ufermann    *
150 ! *                      *
160 ! *   Bearbeitet vom     *
170 ! *     Team des         *
180 ! *  Aktuell Verlages    *
190 ! *      Muenchen        *
200 ! *                      *
210 ! * Benoetigte Geraete   *
220 ! *  TI99/4A  Konsole    *
230 ! *     Ext. Basic       *
240 ! *    Joystick (1)      *
250 ! *  Kassettenrekorder   *
260 ! * Optional:32K-Erw.&   *
262 ! * Speech-Synthesizer   *
264 ! *                      *
270 ! *  Speicherbelegung    *
280 ! *    13369 Bytes       *
290 ! ************************
295 !
300 ! * BEGRUESSUNG-VORSPANN
 *
310 CALL CLEAR :: CALL INIT
:: CALL PEEK(-28672,A)
320 IF A=0 THEN 330 ELSE CAL
L SAY("HOW ARE YOU")
330 CALL HCHAR(1,1,33,768)::
 CALL CLEAR :: CALL VCHAR(1,
1,33,768):: CALL CLEAR
340 IF A=0 THEN 350 ELSE CAL
L SAY("I M A1 COMPUTER,DO YO
U WANT TO PLAY")
350 CALL HCHAR(1,1,63,768)::
 CALL CLEAR :: CALL VCHAR(1,
1,63,768):: IF A=0 THEN
360 ELSE CALL SAY("OK")
360 CALL CLEAR
370 CALL SCREEN(2)
380 PRINT ".........^^^^^^^
^^"
390 PRINT ".........^^^^^^^^
^^^"
400 PRINT ".........^^^^^^^^^
^^^^"
410 PRINT ".......IDEE..&.GR
APHIK"
420 PRINT "......WERNER...UF
ERMANN":
430 PRINT ".....PROGR.JENS.U
FERMANN"
440 PRINT "....COPYRIGHT.BY.
UFERSOFT"
450 Z=12
460 M$=".........VULKANIEN.
"
470 CALL CHAR(139,"",140,"00
0000001",141,"000000281C3808
",142,"2002543834580AA0")
480 FOR I=0 TO 14 :: CALL CO
LOR(I,16,2):: NEXT I
490 CALL SPRITE(#1,139,16,1,
1)
500 FOR N=11 TO 20
510 FOR L=0 TO 10 :: CALL SO
UND(-99,-7,L,120,L,131,L)::
NEXT L
520 C=ASC(SEG$(M$,N,9)):: IF
 C>32 THEN 540
530 FOR D=1 TO 60 :: NEXT D
:: GOTO 580
540 CALL LOCATE(#1,Z*8-7,N*8
+9)
550 CALL MAGNIFY(1):: FOR Q=
0 TO 2 :: CALL PATTERN(#1,Q+
140):: FOR D=0 TO 15 :: NEXT
 D :: NEXT Q
560 CALL LOCATE(#1,Z*8-13,N*
8+4):: CALL MAGNIFY(2):: FOR
 Q=0 TO 1 :: CALL PATTERN(#1
,Q+140):: FOR D=1 TO 4 :: NE
XT D :: NEXT Q
570 CALL PATTERN(#1,142):: C
ALL HCHAR(Z,N+2,C):: CALL PA
TTERN(#1,139)
580 NEXT N :: CALL CLEAR
590 T=350
600 CALL SOUND(T,440,2):: CA
LL SOUND(T/2,554,2):: CALL S
OUND(T/2,659,2):: CALL SOUND
(2*T,880,2)
610 ! * SPIELERKLAERUNG *
620 PRINT "§§>SIE SIND BALLO
NFLIEGER<§§§.............
.........§"
630 PRINT "§ mystische >> wi
nde haben §§ sie in ihrem ba
llon nach §§ vulkanien >> ab
getrieben §"
640 PRINT "§ giftige  wolke
n,  feuer §§ und gluehende l
avasteine §§ bedeuten hoechs
te gefahr §"
650 PRINT "§ die einwohner..
verlassen §§ in grosser eile
 das land §§.............
.........§"
660 PRINT "§ sie haben 5 was
serbomben §§ und sollten die
se in die §"
```

```
670 PRINT "§ vulkan-krater..
einwerfen §§ damit diese  ve
rloeschen §§ zuvor muessen
sie  aber §§ nach rechts > 2
 gefahren §"
680 PRINT "§ zonen durchflie
gen, erst §§ dann koennen si
e die >5< §§ loeschbomben..a
usklinken §§.................
.........§"
690 PRINT "§§§§§§§>BITTE WAR
TEN<§§§§§§§"
700 FOR V=1 TO 12 :: FOR M=1
 TO 30 STEP 2 :: CALL SOUND(
-50,1100,M,2200,M,4400,M)::
CALL SOUND(-50,4400,M,8800,M
,13200,M):: NEXT M :: NEXT V
 :: CALL CLEAR
710 PRINT "§§§§§§§§§§§§§§§§
§§§§§§§§§§§§.................
.........§"
720 PRINT "§ kommen sie aber
 nicht in §§ die naehe der..
schwarzen §§ wolken, > der f
euer oder §§ der lavasteine,
 sogleich §"
730 PRINT "§ werden  sie..wi
eder..zum §§ start >> zuruec
kgeblasen §§.................
.........§"
740 PRINT "§ wenn sie ihre 3
 aufgaben §§ erfuellt haben,
...werden §§ sie zur landung
..gerufen §§ und als..held d
er nation §"
750 PRINT "§ gefeiert!!!....
.........§§.................
.........§§.DEIN.NAME:>....
.....>AEÜ.§§.................
.........§"
760 PRINT "§ alpha-lock-tast
e..loesen §§.................
.........§§§§§§§§§§§§§§§§§§
§§§§§§§§§§"
770 FOR V=1 TO 12 :: FOR J=0
 TO 30 STEP 3 :: CALL SOUND(
-100,3400,J):: CALL SOUND(-2
00,2100,J):: NEXT J :: NEXT
V
780 ACCEPT AT(19,14)SIZE(9)B
EEP VALIDATE(UALPHA):N$
790 CALL KEY(0,K,S):: IF S=0
 THEN 790
800 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL CHARSET
810 !
820 ! *ZEICHENDEFINITION *
830 !
840 W1$="00000000041F1F3F3F7
F3F7F7F3F1F1B00000000C0E8FCF
EFEFCFCFEFCFCF820"
850 B2$="0000000000000009050
707070303010000000000000002
040C0C0C080800000"
860 TI$="070F183D7D7F5520100
8040703030301E0F0B8BCBEFEAA0
4081020E0C0C0C080"
870 MA$="187E2424180098FC1A1
A1A1818242466000000000000000
00000000000000000"
880 FR$="0018255A80BC2759983
C427E0024246600000000000000
00000000000000000"
890 SG$="1140090054981C24284
019727773D8280022148007B6E27
032510402246E6C44"
900 SK$="480084080A204442242
A4A10A4A9545B000000000000000
00000000000000000"
910 F1$="4510A290222A0B41484
81535A5A5A58191542595802A2A2
20849415555515444"
920 F2$="91542595802A2A22084
941555551544481A5A5A53515484
8410B2A2290A21045"
930 FK1$="91542595802A2A2208
4941555551544400000000000000
00000000000000000"
940 FK2$="4510A290222A0B4148
481535A5A5A58100000000000000
00000000000000000"
950 S1$="7F79793E3F9C2E47891
2448912204488FE9E9E7CFC3974E
291482291480422 11"
960 S2$="7F797FBE3F1D0E47011
2400802200480FE9EFE7DFCB870E
28048021040042001"
970 O$="3F1F0F4763717B7E7E7B
7163470F1F3FFCF8F0E2C68EDE7E
7EDE8EC6E2F0F8FC"
980 CALL SCREEN(14)
990 CALL CHAR(92,W1$,132,B2$
,100,TI$,108,F1$,116,F2$,124
,FK1$,128,FK2$,36,MA$,44,FR$
,52,SG$,68,SK$)
1000 CALL CHAR(88,"41545A2A2
A3A0A4A",89,"4C2818101010101
0")
1010 CALL CHAR(97,RPT$("F",1
6),98,"0103070F1F3F7FFF",99,
"80C0E0F0F8FCFEFF",104,RPT$(
"F",16))
```

```
1020 CALL CHAR(105,"0103070F
1F3F7FFF",106,"80C0E0F0F8FCF
EFF",107,"FF7F3F1F0F070301")
1030 CALL CHAR(112,RPT$("F",
16),113,"0103070F1F3F7FFF",1
14,"80C0E0F0F8FCFEFF",120,RP
T$("F",16),121,"0103070F1F3F
7FFF")
1040 CALL CHAR(40,"80C0E0F0F
8FCFEFF",41,"FF7F3F1F0F07030
1",48,"0103070F1F3F7FFF",49,
"80C0E0F0F8FCFEFF",122,"80C0
E0F0F8FCFEFF")
1050 CALL CHAR(58,"80C0E0F0F
8FCFEFF",59,"0103070F1F3F7FF
F",60,"FFFEFCF8F0E0C080",61,
"FF7F3F1F0F070301",64,"01030
70F1F3F7FFF")
1060 ! *BILDAUFBAU*
1070 CALL SCREEN(14):: CALL
MAGNIFY(3)
1080 A$(1)="................
................................
................................
................................
............"
1090 A$(2)="................
............aac.............
............aaac..qr........
............aaaacqppr.......
..........."
1100 A$(3)="aaaaa(ppprihhj..
qpr.........aaaaaa(pp;hhhhjq
pppr..bc....aaaaaaa(;hhhhhh:
pppprbaacihhaaaaaaa§hhhhhhhh
:pppp)aa§hhh"
1110 A$(4)="aaaaaa§hhhhhhhhh
h:pppp)§hhhhaaaaa§hhhh<pp=hh
hh:ppp;hhhhh....ihhhhhpppphh
hhh:p;hhhhhh...ihhhhhh0xx1hh
hhhhhhhhhhhh"
1120 A$(5)=".X.......yxxxxz.
..ihhhhhhhhh.Y..X...yxxxxxxz
...X....X..k....Y..yxxxxxxxx
z..Y....Y..."
1130 A$(6)="xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxx"
1140 CALL COLOR(2,15,2,3,12,
2,4,16,2,5,16,15,6,2,16,7,2,
16,8,2,14)
1150 CALL COLOR(9,15,14,10,1
6,14,11,2,14,12,12,14)
1160 DISPLAY AT(1,1):A$(1);A
$(2);A$(3);A$(4);A$(5);A$(6)
1170 CALL HCHAR(7,1,98):: CA
LL HCHAR(6,2,98):: CALL VCHA
R(8,1,97,7):: CALL VCHAR(7,2
,97,8):: CALL VCHAR(20,1,120
,3):: CALL VCHAR(20,2,120,3)
1180 CALL HCHAR(11,31,106)::
 CALL HCHAR(12,32,106):: CAL
L VCHAR(12,31,104,7):: CALL
VCHAR(13,32,104,6)
1190 CALL VCHAR(20,31,120,3)
:: CALL VCHAR(20,32,120,3)
1200 CALL SPRITE(#16,36,2,16
0,5,0,3,#17,44,2,160,15,0,3)
!MANN/FRAU
1210 CALL SPRITE(#22,36,2,16
0,35,0,3,#23,44,2,160,45,0,3
)
1220 CALL SPRITE(#1,108,11,2
4,16,#2,108,9,48,104,#3,124,
11,48,152,#4,108,12,64,224,#
7,124,10,104,176)!FEUER
1230 CALL SPRITE(#5,92,2,5,1
00,0,2,#9,92,2,15,120,0,1.8,
#10,92,2,30,130,0,2.3,#12,92
,2,45,170,0,3.8)!WOLKEN
1240 !*SPIELABLAUF*
1250 B2=1
1260 FOR I=0 TO 30 :: CALL S
OUND(100,-6,I):: NEXT I
1270 CALL SOUND(100,-7,0)::
CALL SPRITE(#6,100,11,7,18)
1280 IF B2=2 THEN 1290 ELSE
1300
1290 IF B2=1 THEN 1300 :: CA
LL SPRITE(#18,52,10,48,104,-
8,0,#19,68,10,48,152,-12,0)!
LAVA
1300 CALL JOYST(1,X,Y):: CAL
L MOTION(#6,-Y*2,X*2,#5,0,-8
,#9,0,7,#10,0,-8,#11,0,7,#12
,0,-7)
1310 CALL PATTERN(#1,116,#2,
116,#3,128,#4,116,#7,128)
1320 CALL PATTERN(#1,108,#2,
108,#3,124,#4,108,#7,124)
1330 CALL POSITION(#6,X,Y)::
 IF (X<7 OR X>56)OR(Y<18)THE
N CALL DELSPRITE(#6):: CALL
SOUND(50,300,0):: GOTO 1270
1340 IF Y>210 THEN 1480
1350 CALL COINC(#6,#5,20,T1)
:: IF T1=-1 THEN 1270 ELSE 1
360
1360 CALL COINC(#6,#9,17,T2)
:: IF T2=-1 THEN 1270 ELSE 1
370
```

```
1370 CALL COINC(#6,#10,17,T3
):: IF T3=-1 THEN 1270 ELSE
1380
1380 CALL COINC(#6,#12,17,T4
):: IF T4=-1 THEN 1270 ELSE
1390
1390 CALL COINC(#6,#1,10,T5)
:: IF T5=-1 THEN 1270 ELSE 1
400
1400 CALL JOYST(1,X,Y):: CAL
L MOTION(#6,-Y*2,X*2):: CALL
 PATTERN(#1,116,#2,116,#3,12
8,#4,116,#7,128)
1410 CALL COINC(#6,#2,17,T6)
:: IF T6=-1 THEN 1270 ELSE 1
420
1420 CALL COINC(#6,#3,17,T7)
:: IF T7=-1 THEN 1270 ELSE 1
430
1430 CALL JOYST(1,X,Y):: CAL
L MOTION(#6,-Y*2,X*2):: CALL
 PATTERN(#1,108,#2,108,#3,12
4,#4,108,#7,124)
1440 CALL COINC(#6,#4,17,T8)
:: IF T8=-1 THEN 1270 ELSE 1
450
1450 CALL COINC(#6,#18,12,M1
):: IF M1=-1 THEN 1270 ELSE
1460
1460 CALL COINC(#6,#19,12,M2
):: IF M2=-1 THEN 1270 ELSE
1470
1470 GOTO 1290
1480 CALL DELSPRITE(#6):: B2
=B2+1 :: IF B2=2 THEN 1260 E
LSE 1490
1490 CALL DELSPRITE(#18,#19)
:: FOR I=0 TO 30 :: CALL SOU
ND(100,-7,I):: NEXT I :: CAL
L LOCATE(#6,7,15)
1500 P=0 :: Z=5 :: CALL COLO
R(#5,15,#9,15,#10,15,#12,15)
1510 CALL MOTION(#6,-0,3)::
IF P=50 THEN 1710 :: IF Z=0
THEN 2430
1520 CALL SOUND(200,-4,0)::
CALL KEY(1,K,S):: CALL PATTE
RN(#1,108,#2,108,#3,124,#4,1
08,#7,124)
1530 CALL PATTERN(#1,116,#2,
116,#3,128,#4,116,#7,128)
1540 IF S=0 THEN 1520
1550 Z=Z-1
1560 CALL POSITION(#6,Y1,X1)
:: CALL SPRITE(#8,132,5,Y1,X
1,4,2):: CALL SOUND(-90,523,
2)
1570 CALL PATTERN(#1,108,#2,
108,#3,124,#4,108,#7,124)::
CALL COINC(#8,#1,8,A):: IF A
=-1 THEN 1580 ELSE 1590
1580 P=P+10 :: CALL DELSPRIT
E(#8,#1):: CALL SOUND(200,-5
,0,800,0):: GOTO 1510
1590 CALL COINC(#8,#2,8,B)::
 IF B=-1 THEN 1600 ELSE 1610
1600 P=P+10 :: CALL DELSPRIT
E(#2,#8):: CALL SOUND(200,-5
,0,800,0):: GOTO 1510
1610 CALL SOUND(100,3000,0):
: CALL COINC(#8,#3,5,C):: IF
 C=-1 THEN 1620 ELSE 1630
1620 P=P+10 :: CALL DELSPRIT
E(#3,#8):: CALL SOUND(200,-5
,0,800,0):: GOTO 1510
1630 CALL PATTERN(#1,116,#2,
116,#3,128,#4,116,#7,128)::
CALL COINC(#4,#8,8,D):: IF D
=-1 THEN 1640 ELSE 1650
1640 P=P+10 :: CALL DELSPRIT
E(#4,#8):: CALL SOUND(200,-5
,0,800,0):: GOTO 1510
1650 CALL COINC(#8,#7,5,E)::
 IF E=-1 THEN 1660 ELSE 1670
1660 P=P+10 :: CALL DELSPRIT
E(#8,#7):: CALL SOUND(200,-5
,0,800,0):: GOTO 1510
1670 CALL POSITION(#8,YPOS,X
POS):: IF XPOS>230 OR YPOS>1
50 THEN CALL DELSPRITE(#8)::
 GOTO 1510
1680 IF YPOS>=150 THEN 1510
1690 IF P=50 THEN 1710 ELSE
1570
1700 ! *SPIEL GESCHAFFT*
1710 CALL DELSPRITE(#8):: CA
LL SCREEN(5)
1720 CALL COLOR(8,4,5,9,15,5
,10,16,5,11,2,5,12,12,5)
1730 CALL COLOR(#5,16,#9,16,
#10,16,#12,16):: CALL MOTION
(#5,0,2,#9,0,1.8,#10,0,2.3,#
12,0,3.8)
1740 CALL LOCATE(#16,160,85,
#17,160,100,#22,160,120,#23,
160,135):: CALL MOTION(#16,0
,0,#17,0,0,#22,0,0,#23,0,0)
1750 CALL CHAR(136,S1$,140,S
2$)
1760 CALL SPRITE(#15,136,11,
1,190)
1770 CALL CHAR(88,"14082A1C3
61C2A08",89,"080A0C281808081
4")
```

```
1780 CALL POSITION(#6,X1,Y1)
:: CALL SOUND(10,233,9)
1790 IF Y1=104 THEN CALL MOT
ION(#6,3,0)ELSE 1780
1800 T=400 :: CALL POSITION(
#6,X2,Y2):: T=T-2 :: CALL SO
UND(-5,T,0):: IF X2=105 THEN
 CALL MOTION(#6,0,0)ELSE 180
0
1810 T2=400
1820 T4=T2/2
1830 T8=T2/4
1840 L=0
1850 GOSUB 1950
1860 L=10
1870 GOSUB 1950
1880 L=2
1890 GOSUB 2140
1900 L=5
1910 GOSUB 1950
1920 GOTO 2340
1930 ! SATZ A
1940 !
1950 FOR J=1 TO 2
1960 CALL SOUND(T4,784,L+1,2
62,L+3)
1970 CALL SOUND(T4,784,L+3,3
92,L+6)
1980 CALL SOUND(T8,698,L+3,2
62,L+6)
1990 CALL SOUND(T8,659,L+3,2
62,L+6)
2000 CALL SOUND(T8,587,L+3,3
92,L+6)
2010 CALL SOUND(T8,523,L+3,3
92,L+6)
2020 NEXT J
2030 CALL SOUND(T8,530,L+1)
2040 CALL SOUND(T8,349,L+3)
2050 CALL SOUND(T4,392,L+3)
2060 CALL SOUND(T4,349,L+3)
2070 CALL SOUND(T4,294,L)
2080 CALL SOUND(T4,392,L+3)
2090 CALL SOUND(T2,262,L+2)
2100 RETURN
2110 !
2120 ! SATZ B
2130 !
2140 CALL SOUND(T8,494,L+1,1
96,L+3)
2150 CALL SOUND(T8,525,L+3,1
96,L+6)
2160 CALL SOUND(T4,587,L+2,3
92,L+6)
2170 CALL SOUND(T8,494,L+2,1
96,L+6)
2180 CALL SOUND(T8,523,L+3,1
96,L+6)
2190 CALL SOUND(T4,587,L+2,3
92,L+6)
2200 CALL SOUND(T4,784,L+1,1
96,L+3)
2210 CALL SOUND(T4,587,L+3,3
92,L+6)
2220 CALL SOUND(T4,587,L+3,1
96,L+3)
2230 CALL SOUND(T4,587,L+3,3
92,L+6)
2240 CALL SOUND(T4,784,L+1,1
96,L+3)
2250 CALL SOUND(T4,587,L+3,3
92,L+6)
2260 CALL SOUND(T4,784,L+3,1
96,L+6)
2270 CALL SOUND(T4,587,L+3,3
92,L+6)
2280 CALL SOUND(T8,523,L+1,1
96,L+3)
2290 CALL SOUND(T8,484,L+3,1
96,L+6)
2300 CALL SOUND(T8,440,L+3,1
96,L+6)
2310 CALL SOUND(T8,392,L+3,1
96,L+6)
2320 CALL SOUND(T2,330,L+2,2
62,L+4)
2330 RETURN
2340 IF A=0 THEN 2350 ELSE C
ALL SAY("THAT IS RIGHT AND V
ERY GOOD, GOOD WORK, GOOD BY
E")
2350 FOR I=1 TO 3 :: CALL SO
UND(200,261,0,329,0,391,0)::
 CALL SOUND(300,523,0,659,0,
783,0):: NEXT I
2360 CALL CLEAR :: CALL DELS
PRITE(#1,#2,#3,#4,#5,#7,#9,#
10,#12):: CALL SCREEN(2):: C
ALL CHARSET :: CALL MAGNIFY(
4)
2370 FOR Z=1 TO 8 :: CALL CO
LOR(Z,16,2):: NEXT Z
2380 CALL CHAR(108,0$):: CAL
L SPRITE(#20,108,16,100,165)
:: DISPLAY AT(6,2)SIZE(10):N
$ :: DISPLAY AT(6,22)SIZE(5)
:"TASTE"
2390 DISPLAY AT(8,2):"WIR VE
RLEIHEN..............DIR DE
N........HELDENORDEN........
.........^^^^^^^^^^"
```

```
2400 CALL MAGNIFY(4):: DISPL
AY AT(15,2)SIZE(9):"GAME OVE
R" :: DISPLAY AT(6,22)SIZE(5
):"TASTE"
2410 CALL KEY(1,K,S):: CALL
PATTERN(#1,108,#2,108,#3,124
,#4,108,#7,124)
2420 CALL PATTERN(#15,136,#1
,116,#2,116,#3,128,#4,116,#7
,128,#15,140):: IF S=0 THEN
2410 ELSE 800
2430 CALL DELSPRITE(#8)
2440 CALL SCREEN(7)!DUNKELRO
T
2450 CALL COLOR(8,2,7,8,15,7
,10,16,7,11,2,7,12,12,7)
2460 FOR J=30 TO 0 STEP -1 :
: CALL SOUND(100,-7,J):: NEX
T J
2470 FOR I=0 TO 30 :: CALL S
OUND(50,-6,I):: NEXT I
2480 IF A=0 THEN 2490 ELSE C
ALL SAY("HELP,HELP,HELP,WHAT
 IS THAT,I M SORRY")
2490 CALL CLEAR :: CALL DELS
PRITE(#1,#2,#3,#4,#7,#5,#9,#
10,#11,#12,#13):: CALL CHARS
ET
2500 CALL SCREEN(2):: FOR F=
1 TO 8 :: CALL COLOR(F,16,2)
:: NEXT F
2510 CALL MAGNIFY(4):: DISPL
AY AT(18,2)SIZE(10):N$ :: DI
SPLAY AT(20,2)SIZE(25):"BITT
E WEITER UEBEN !!!" :: DISPL
AY AT(6,22)SIZE(5):"TASTE"
2520 CALL KEY(1,K,S)
2530 IF S=0 THEN 2510 ELSE 8
00
2540 GOTO 2360
```

Fortsetzung von Seite 54

ersten Runden hast Du jedoch Deinen ganzen Sand verbracht und Dein Ballon ist steuerlos geworden. Trotzdem mußt Du versuchen, beim Überfliegen der Krater Deine Wasserbomben genau zu plazieren, denn Dir stehen für 5 brennende Vulkane ja nur 5 Wasserbomben zur Verfügung. Wirfst Du auch nur eine vorbei, ist das Land Vulkanien rettungslos verloren und Du mußt in einem neuen Spiel noch einmal Deinen Mut beweisen.
Sollte es Dir aber gelingen, alle Vulkane zu löschen, wirst Du von den glücklichen Einwohnern zur Landung aufgefordert und für Deine Leistungen gebührend geehrt.

Werner Ufermann

BLUMEN GIESSEN

Über die Pfeiltasten der Tastatur bewegst Du einen Zwerg, der Blumen vor dem Verwelken retten muß. Steuere den Zwerg ein Feld rechts neben die verwelkte Blume und gieße sie (Taste 'Q'). Dafür erhälst Du 10 Punkte. Gießt Du aber eine Blume, die noch nicht verwelkt ist, werden Dir 10 Punkte abgezogen. Sind 5 Blumen auf einmal verwelkt, stirbt der Zwerg an Sauerstoffmangel und das Spiel ist beendet.
Auf der unteren Etage des Gewächshauses wird der Zwerg von einer Maus bei seiner Arbeit behindert, in den oberen Etagen von einer Spinne. Nur durch Überspringen nach rechts (Taste '.') oder links (Taste ',') kann der Zwerg sein Leben retten und seine Arbeit fortsetzen. Für einen erfolgreichen Sprung erhälst Du außerdem noch 20 Punkte. Beim Springen darf weder eine der Leitern noch einer der Blumentöpfe berührt werden.

Gunther Krämling

```
100 REM**********************
110 REM*                    *
120 REM* BLUMEN GIESSEN     *
130 REM*  Copyright by      *
140 REM*Gunther  Kraemling*
150 REM*                    *
160 REM* Bearbeitet  vom    *
170 REM*     Team  des      *
180 REM* Aktuell Verlages   *
190 REM*      Muenchen      *
200 REM*                    *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole   *
230 REM*                    *
270 REM* Speicherbelegung   *
280 REM*    7015 Bytes      *
290 REM**********************
295 REM
300 CALL CLEAR
310 CALL CHAR(40,"0007653F17
1717")
320 CALL CHAR(78,"187E181818
3C7EFF")
330 CALL CHAR(37,"FFFFFFFFFF
FFFF")
340 CALL CHAR(64,"00000000FF
FFFF")
350 CALL CHAR(65,"7E5A7E18FF
182442")
360 CALL CHAR(66,"0808080808
080808")
370 CALL CHAR(67,"0000000001
071F3F")
380 CALL CHAR(68,"0000000080
E0F8FC")
390 CALL CHAR(69,"7F7F7F7F7F
7F7F7F")
400 CALL CHAR(70,"FEFEFEFEFE
FEFEFE")
410 CALL CHAR(71,"FFFFFF")
420 CALL CHAR(35,"0014081408
0A0C08")
430 CALL CHAR(36,"0000007848
A84CAA")
440 CALL CHAR(72,"7E5AFF99A5
A5")
450 CALL CHAR(73,"00007E5AFF
99A5A5")
460 CALL CHAR(74,"04020D3140
81020C")
470 CALL CHAR(75,"9A494C2299
442310")
480 CALL CHAR(76,"1020C00003
0C1020")
490 CALL CHAR(77,"102C438404
020202")
500 CALL CHAR(124,"000004022
16D7E7E")
510 CALL CHAR(106,"FFFFFFFFF
FFFFFFF")
520 CALL CHAR(80,"7E3C3C3C3C
3C1818")
530 CALL CHAR(88,"FFFFFF")
540 CALL CHAR(96,"243C243C24
3C243C")
550 CALL COLOR(1,4,1)
560 CALL COLOR(2,8,1)
570 CALL COLOR(3,5,1)
580 CALL COLOR(4,5,1)
590 CALL COLOR(5,2,1)
600 CALL COLOR(6,16,1)
610 CALL COLOR(7,5,1)
620 CALL COLOR(8,10,1)
630 CALL COLOR(9,11,1)
640 CALL COLOR(10,14,1)
650 CALL COLOR(11,14,1)
660 CALL COLOR(12,15,1)
670 CALL SCREEN(7)
680 CALL CLEAR
690 B=0
700 P=0
710 GOSUB 3350
720 GOSUB 3460
730 G=0
740 X=19
750 Y=3
760 CALL HCHAR(1,1,64,32)
770 CALL HCHAR(21,1,71,32)
780 CALL HCHAR(3,1,106,32)
790 CALL HCHAR(20,1,37,32)
800 CALL VCHAR(4,32,106,16)
810 CALL VCHAR(4,1,106,16)
820 CALL VCHAR(4,29,106,3)
830 CALL HCHAR(6,30,106,2)
840 CALL HCHAR(4,30,74)
850 CALL HCHAR(4,31,75)
860 CALL HCHAR(5,30,76)
870 CALL HCHAR(5,31,77)
880 CALL HCHAR(18,29,67)
890 CALL HCHAR(18,30,68)
900 CALL HCHAR(19,29,69)
910 CALL HCHAR(19,30,70)
920 CALL VCHAR(15,7,96,4)
930 CALL VCHAR(15,20,96,4)
940 CALL HCHAR(15,8,88,12)
950 CALL VCHAR(10,8,96,4)
960 CALL VCHAR(10,19,96,4)
970 CALL HCHAR(10,9,88,10)
980 FOR I=12 TO 15 STEP 3
990 CALL HCHAR(7,I,35)
1000 CALL HCHAR(8,I,80)
```

```
1010 NEXT I
1020 FOR I=10 TO 12 STEP 2
1030 CALL HCHAR(12,I,35)
1040 CALL HCHAR(13,I,80)
1050 NEXT I
1060 FOR I=15 TO 17 STEP 2
1070 CALL HCHAR(12,I,35)
1080 CALL HCHAR(13,I,80)
1090 NEXT I
1100 FOR I=9 TO 18 STEP 3
1110 CALL HCHAR(17,I,35)
1120 CALL HCHAR(18,I,80)
1130 NEXT I
1140 CALL HCHAR(X,Y,65)
1150 CALL KEY(0,K,S)
1160 IF K<>68 THEN 1250
1170 CALL GCHAR(X+1,Y+1,A)
1180 IF A=32 THEN 1560
1190 IF Y=27 THEN 1560
1200 CALL HCHAR(X,Y,32)
1210 Y=Y+1
1220 G=G+1
1230 CALL SOUND(1,800,0)
1240 GOTO 1560
1250 IF K<>83 THEN 1340
1260 CALL GCHAR(X+1,Y-1,A)
1270 IF A=32 THEN 1560
1280 IF Y=2 THEN 1560
1290 CALL HCHAR(X,Y,32)
1300 Y=Y-1
1310 G=G+1
1320 CALL SOUND(1,800,0)
1330 GOTO 1560
1340 IF K<>69 THEN 1450
1350 CALL HCHAR(X,Y,32)
1360 CALL GCHAR(X-1,Y,A)
1370 IF A<>96 THEN 1450
1380 X=X-5
1390 G=G+1
1400 FOR I=-1 TO -4 STEP -1
1410 CALL SOUND(100,I,0)
1420 NEXT I
1430 IF (X=14)-(X=9)THEN 238
0
1440 GOTO 1560
1450 IF K<>88 THEN 1560
1460 CALL HCHAR(X,Y,32)
1470 CALL GCHAR(X+1,Y,A)
1480 IF A<>96 THEN 1560
1490 X=X+5
1500 G=G+1
1510 FOR I=-4 TO -1
1520 CALL SOUND(100,I,0)
1530 NEXT I
1540 IF X=19 THEN 2000
1550 IF X=14 THEN 2380
1560 CALL HCHAR(X,Y,65)
1570 IF K<>81 THEN 1700
1580 CALL HCHAR(X,Y-1,40)
1590 CALL GCHAR(X-2,Y-1,A)
1600 IF A=36 THEN 1640
1610 P=P-10
1620 GOSUB 3350
1630 GOTO 1690
1640 B=B-1
1650 CALL SOUND(200,-5,0)
1660 CALL HCHAR(X-2,Y-1,35)
1670 P=P+10
1680 GOSUB 3350
1690 CALL HCHAR(X,Y-1,32)
1700 RANDOMIZE
1710 V=INT(RND*11)+7
1720 IF (V<>7)*(V<>12)*(V<>1
7)THEN 1710
1730 W=INT(RND*18)+9
1740 IF (W=11)*(W=13)*(W=14)
*(W=16)THEN 1730
1750 CALL GCHAR(V,W,A)
1760 IF A<>35 THEN 1800
1770 CALL HCHAR(V,W,36)
1780 B=B+1
1790 IF B=5 THEN 1840
1800 IF G<10 THEN 1140
1810 G=0
1820 IF X=19 THEN 2000
1830 GOTO 2380
1840 CALL HCHAR(X,Y,78)
1850 FOR I=1 TO 2
1860 CALL SOUND(500,660,0,67
0,0,680,0)
1870 CALL SOUND(500,560,0,57
0,0,560,0)
1880 CALL SOUND(500,680,0,67
0,0,650,0)
1890 CALL SOUND(500,580,0,57
0,0,560,0)
1900 NEXT I
1910 IF P<=HP THEN 1940
1920 HP=P
1930 GOSUB 3460
1940 FOR I=1 TO 350
1950 CALL KEY(0,K,S)
1960 IF S<>0 THEN 680
1970 NEXT I
1980 CALL CLEAR
1990 END
2000 M=19
2010 N=28
2020 CALL GCHAR(M,N,A)
2030 IF A=65 THEN 1840
```

```
2040 CALL HCHAR(M,N,124)
2050 CALL KEY(0,K,S)
2060 IF K<>83 THEN 2120
2070 IF Y-1=2 THEN 2120
2080 CALL HCHAR(X,Y,32)
2090 Y=Y-1
2100 CALL SOUND(1,800,0)
2110 GOTO 2340
2120 IF K<>68 THEN 2190
2130 CALL GCHAR(X,Y+1,A)
2140 IF A=124 THEN 1840
2150 CALL HCHAR(X,Y,32)
2160 Y=Y+1
2170 CALL SOUND(1,800,0)
2180 GOTO 2340
2190 IF K<>46 THEN 2340
2200 CALL GCHAR(X-1,Y+1,A)
2210 IF A<>32 THEN 1840
2220 CALL HCHAR(X,Y,32)
2230 CALL HCHAR(X-1,Y+1,65)
2240 CALL SOUND(-1,120,0)
2250 CALL HCHAR(X-1,Y+1,32)
2260 CALL GCHAR(X,Y+1,A)
2270 IF A<>124 THEN 2330
2280 CALL HCHAR(M,N,32)
2290 Y=Y+2
2300 P=P+20
2310 GOSUB 3350
2320 GOTO 1140
2330 Y=Y+2
2340 CALL HCHAR(X,Y,65)
2350 CALL HCHAR(M,N,32)
2360 N=N-1
2370 GOTO 2020
2380 CALL HCHAR(X,Y,65)
2390 IF Y<=8 THEN 2420
2400 T=8
2410 GOTO 2430
2420 T=19
2430 U=4
2440 FOR I=4 TO 8
2450 CALL HCHAR(U,T,32)
2460 U=I+1
2470 CALL HCHAR(I,T,66)
2480 CALL HCHAR(U,T,72)
2490 NEXT I
2500 CALL HCHAR(U,T,73)
2510 FOR I=8 TO 4 STEP -1
2520 CALL HCHAR(I,T,32)
2530 NEXT I
2540 IF X<>14 THEN 2570
2550 CALL HCHAR(U,T,32)
2560 U=14
2570 CALL HCHAR(U,T,73)
2580 CALL GCHAR(X,Y,A)
2590 IF A=73 THEN 1840
2600 CALL KEY(0,K,S)
2610 IF Y<T THEN 2940
2620 IF S=0 THEN 3290
2630 IF K<>83 THEN 2770
2640 CALL GCHAR(X,Y-1,A)
2650 IF A=73 THEN 1840
2660 CALL HCHAR(X,Y,32)
2670 Y=Y-1
2680 CALL SOUND(1,800,0)
2690 GOTO 3290
2700 IF K<>68 THEN 2770
2710 CALL GCHAR(X+1,Y+1,A)
2720 IF A=32 THEN 2770
2730 CALL HCHAR(X,Y,32)
2740 Y=Y+1
2750 CALL SOUND(1,800,0)
2760 GOTO 3290
2770 IF K<>44 THEN 3290
2780 CALL GCHAR(X-1,Y-1,A)
2790 IF A<>32 THEN 1840
2800 CALL HCHAR(X,Y,32)
2810 CALL HCHAR(X-1,Y-1,65)
2820 CALL SOUND(-1,120,0)
2830 CALL HCHAR(X-1,Y-1,32)
2840 CALL GCHAR(X,Y-1,A)
2850 IF A<>73 THEN 2910
2860 CALL HCHAR(U,T,32)
2870 Y=Y-2
2880 P=P+20
2890 GOSUB 3350
2900 GOTO 1140
2910 Y=Y-2
2920 GOTO 3290
2930 IF S=0 THEN 3230
2940 IF K<>83 THEN 3010
2950 CALL GCHAR(X+1,Y-1,A)
2960 IF A=32 THEN 3010
2970 CALL HCHAR(X,Y,32)
2980 Y=Y-1
2990 CALL SOUND(1,800,0)
3000 GOTO 3230
3010 IF K<>68 THEN 3080
3020 CALL GCHAR(X,Y+1,A)
3030 IF A=73 THEN 1840
3040 CALL HCHAR(X,Y,32)
3050 Y=Y+1
3060 CALL SOUND(1,800,0)
3070 GOTO 3230
3080 IF K<>46 THEN 3230
3090 CALL GCHAR(X-1,Y+1,A)
3100 IF A<>32 THEN 1840
3110 CALL HCHAR(X,Y,32)
3120 CALL HCHAR(X-1,Y+1,65)
3130 CALL SOUND(-1,120,0)
```

```
3140 CALL HCHAR(X-1,Y+1,32)
3150 CALL GCHAR(X,Y+1,A)
3160 IF A<>73 THEN 3220
3170 CALL HCHAR(U,T,32)
3180 Y=Y+2
3190 P=P+20
3200 GOSUB 3350
3210 GOTO 1140
3220 Y=Y+1
3230 CALL HCHAR(X,Y,65)
3240 CALL GCHAR(X,Y,A)
3250 IF A=73 THEN 1840
3260 CALL HCHAR(U,T,32)
3270 T=T-1
3280 GOTO 2570
3290 CALL HCHAR(X,Y,65)
3300 CALL GCHAR(X,Y,A)
3310 IF A=73 THEN 1840
3320 CALL HCHAR(U,T,32)
3330 T=T+1
3340 GOTO 2570
3350 IF P>0 THEN 3370
3360 P=0
3370 P$="score:......"&STR$(
P)&""
3380 C=22
3390 D=4
3400 FOR I=1 TO LEN(P$)
3410 Z=ASC(SEG$(P$,I,1))
3420 CALL HCHAR(C,D,Z)
3430 D=D+1
3440 NEXT I
3450 RETURN
3460 HP$="highscore:.."&STR$
(HP)&""
3470 E=23
3480 F=4
3490 FOR I=1 TO LEN(HP$)
3500 L=ASC(SEG$(HP$,I,1))
3510 CALL HCHAR(E,F,L)
3520 F=F+1
3530 NEXT I
3540 RETURN
```

BLACK HOLE

Black Hole ist ein Spiel in 4 Bildern für die Grundversion des TI99/4A.
Nach dem Starten des Programmes erscheint das Titelbild, begleitet von der Filmmelodie 'Buck Rogers'. Nähere Erläuterungen zum Ablauf des Spieles gibt der Computer nach Eingabe von 'J'. Im anderen Fall beginnt das Spiel sofort mit Bild 1.
Hier ist mit einem Raumschiff ein enger Korridor zu durchfliegen. Berührt man dabei die Wände, so ist das Spiel damit beendet, kann man jedoch den einzigen Stern in diesem Korridor überfliegen, gelangt man damit zu Bild 2.
Von einem Flugzeug aus springst Du nun mit einem Fallschirm mitten in einen Sumpf, nur ein roter Stein bietet Dir genügend Halt, um zu landen. Dieser Stein ist zugleich auch der Eingang zu Bild 3.
Ein riesiges Labyrinth mit tödlichen Fallen erwartet Dich. Erreichst Du den Ausgang auf der anderen Seite, ohne in eine der 15 unsichtbaren Zeitfallen oder in eines der Energienetze gelaufen zu sein, darfst Du in Bild 4 in den Weltraum starten. Im Labyrinth wimmelt es von Vögeln. Je mehr Du von Ihnen mitnehmen kannst, desto höher steigt Dein Punktekonto. Im Bild 4 siehst Du nun am Monitor Deines Raumkreuzers ein schwarzes Loch, welches ständig die Sterne seiner Umgebung in sich einsaugt. Dieses 'Loch' kannst Du nun dazu verwenden, um feindliche Raumschiffe zu zerstören. Achte aber dabei darauf, daß Du keine Energiekreuze mit einsaugst oder den Bildschirm verläßt, denn dann ist Deine Mission gescheitert und Du mußt es noch einmal von vorne versuchen.
P.S. Schwarze Löcher kann man nur daran erkennen, daß die Sterne der näheren Umgebung verlöschen.
Gesteuert werden alle Funktionen über die Pfeiltasten der Konsole. (FCTN nicht gedrückt).

Stefan Berger

```
100 REM**********************
110 REM*                    *
120 REM*   BLACK HOLE       *
130 REM*  Copyright by      *
140 REM* Stefan  Berger     *
150 REM*                    *
160 REM* Bearbeitet  vom    *
170 REM*    Team  des       *
180 REM* Aktuell Verlages    *
190 REM*     Muenchen       *
200 REM*                    *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole   *
230 REM*                    *
270 REM* Speicherbelegung   *
280 REM*    9576 Bytes      *
290 REM**********************
295 REM
300 RESTORE
310 CALL CLEAR
320 CALL HCHAR(1,1,42,768)
330 READ A1,B1,C1
340 IF A1=0 THEN 370
350 CALL VCHAR(A1,B1,48,C1)
360 GOTO 330
370 DATA 4,3,7,5,6,2,8,6,2,4
,9,7,7,14,4,5,15,2,4,16,2,5,
17,2,7,18,4
380 DATA 5,21,5,4,26,7,13,6,
9,13,10,9,14,13,7,14,17,7,13
,20,9,13,25,9,0,0,0
390 READ A1,B1,C1
400 IF A1=0 THEN 430
410 CALL HCHAR(A1,B1,48,C1)
420 GOTO 390
430 DATA 4,4,2,7,4,2,10,4,2,
10,10,2,8,15,3,4,22,2,10,22,
2,7,27,1,6,28,1,5,29,1,4,30,
1
440 DATA 8,28,1,9,29,1,10,30
,1,17,7,3,13,14,3,21,14,3,21
,21,3,13,26,3,17,26,2,21,26,
3,0,0,0
450 GOSUB 3650
460 CALL CLEAR
470 INPUT "SPIELERLAEUTERUNG
EN?(J/N)":FG$
480 IF (FG$="N")+(FG$="n")TH
EN 710
490 CALL CLEAR
500 PRINT "HALLO,SPIELER!":
: :"DU MUSST DIE GALAXIS RET
TEN!": : :"ZUERST FLIEGE DUR
CH EINEN": :"EINFLUGKORRIDOR
 BIS";
510 PRINT " DU EINEN": :"STE
RN SIEHST.DIESEN MUSST DU":
:"ZERSTOEREN": : : :
520 FOR AS=1 TO 2500
530 NEXT AS
540 CALL CLEAR
550 PRINT "JETZT SPRINGST DU
 MIT EINEM": :"FALLSCHIRM AB
.DU MUSST AUF": :"EINEM ROTE
N STEIN LANDEN": :"(WENN NIC
HT";
560 PRINT "=PUNKTABZUG)": :"
ER IST DER EINGANG ZU EINEM"
: :"LABYRINT!": :
570 FOR AS=1 TO 3000
580 NEXT AS
590 CALL CLEAR
600 PRINT "HIER MUSST DU DAS
 SCHOTT AM": :"ANDEREN ENDE
ERREICHEN!": :"FRESSE DABEI
MOEGLICHST VIEL": :"VOEGEL(=
P";
610 PRINT "UNKTGEWINN)": :"A
BER HUETE DICH IN EIN NETZ":
 :"ODER IN EINE DER 15 UNSIC
HT-": :"BAREN FALLEN ZU LAU"
;
620 PRINT "FEN!": :"DU HAST
3 LEBEN!"
630 FOR AS=1 TO 3500
640 NEXT AS
650 CALL CLEAR
660 PRINT "HINTER DEM SCHOTT
 BEFINDET": :"SICH EIN MONIT
OR,VON DEM AUS": :"DU EIN SC
HWARZES LOCH MANI-": :"PULIE
";
670 PRINT "REN KANNST!": :"D
U SIEHST ES NICHT,MUSST": :"
ABER DAMIT FEINDLICHE RAUM-"
: :"SCHIFFE ZERSTOEREN!": :"
HUETE";
680 PRINT " DICH DAVOR DIE G
ALAXIS": :"(DEN BILDSCHIRM)
ZU VERLAS-": :"SEN ODER AUF
EIN ENERGIENETZ": :"ZU STOSS
EN"
690 FOR AS=1 TO 3500
700 NEXT AS
710 CALL CLEAR
720 PRINT "ACHTUNG!": : :"DA
S SPIEL BEGINNT!": : : : : :
730 FOR AS=1 TO 1000
740 NEXT AS
750 RANDOMIZE
```

```
760 CALL CLEAR
770 CALL CHAR(112,"FFFFFFFFF
FFFFFFF")
780 CALL COLOR(2,7,1)
790 CALL HCHAR(23,1,112,14)
800 CALL VCHAR(1,1,112,23)
810 CALL HCHAR(23,17,112,15)
820 CALL VCHAR(1,32,112,23)
830 X=12
840 A=12
850 X=X+INT(3*RND)-1
860 IF X+4>28 THEN 1050
870 IF X-1<3 THEN 1070
880 PRINT TAB(X-1);CHR$(112)
;CHR$(112);TAB(X+3);CHR$(112
);CHR$(112)
890 SU=SU+1
900 IF SU/50=INT(SU/50)THEN
1140
910 CALL KEY(0,K,S)
920 PU=PU+1
930 IF S=0 THEN 1000
940 IF K=83 THEN 990
950 IF K=68 THEN 970
960 GOTO 1000
970 A=A+1
980 GOTO 1000
990 A=A-1
1000 CALL GCHAR(2,A,G)
1010 CALL HCHAR(2,A,86)
1020 IF G=112 THEN 1090
1030 IF G=42 THEN 1160
1040 GOTO 850
1050 X=X-1
1060 GOTO 880
1070 X=X+1
1080 GOTO 880
1090 CALL CLEAR
1100 CALL SOUND(4000,262,5)
1110 PRINT "DU BIST GEGEN DI
E WAENDE DES": :"EINFLUGKORR
IDORES GEPRALLT!": :
1120 CALL COLOR(2,2,1)
1130 GOTO 2780
1140 CALL HCHAR(23,X+3,42)
1150 GOTO 910
1160 CALL CLEAR
1170 CALL SOUND(200,523,5)
1180 PRINT "ACHTUNG!": :"ABS
PRUNG!": : : : :
1190 FOR AS=1 TO 1000
1200 NEXT AS
1210 CALL CLEAR
1220 CALL SCREEN(8)
1230 RANDOMIZE
1240 CALL CLEAR
1250 REM
1260 CALL HCHAR(24,1,30,32)
1270 CALL CHAR(100,"FFFFFFFF
FFFFFFFF")
1280 CALL CHAR(104,"7EFFC35A
7E181824")
1290 A=2
1300 Y=2*INT(12*RND)+5
1310 CALL COLOR(9,7,1)
1320 CALL COLOR(2,13,1)
1330 CALL HCHAR(24,2*INT(16*
RND)+1,100,1)
1340 CALL KEY(0,K,S)
1350 IF S=0 THEN 1420
1360 IF K=83 THEN 1390
1370 IF K=68 THEN 1410
1380 GOTO 1420
1390 F=-1
1400 GOTO 1420
1410 F=1
1420 IF Y+F=0 THEN 1520
1430 IF Y+F=32 THEN 1550
1440 CALL HCHAR(A-1,Y,32)
1450 Y=Y+F
1460 CALL GCHAR(A+1,Y,G)
1470 CALL HCHAR(A,Y,104)
1480 IF G=100 THEN 1580
1490 IF G=30 THEN 1640
1500 A=A+1
1510 GOTO 1340
1520 CALL HCHAR(A-1,Y,32)
1530 Y=31
1540 GOTO 1450
1550 CALL HCHAR(A-1,Y,32)
1560 Y=1
1570 GOTO 1420
1580 CALL SOUND(200,262,5)
1590 PU=PU+10
1600 FOR AS=1 TO 500
1610 NEXT AS
1620 GOTO 1670
1630 REM FEHLSPRUNG
1640 CALL SOUND(400,523,5)
1650 PU=PU-10
1660 GOTO 1240
1670 CALL CLEAR
1680 CALL COLOR(15,5,1)
1690 CALL CHAR(120,"00183C7E
7E3C18")
1700 CALL COLOR(12,16,1)
1710 CALL CHAR(128,"AA55AA55
AA55AA55")
1720 CALL CHAR(112,"")
1730 CALL CHAR(96,"0F1E3CFFF
F3C1E0F")
```

```
1740 CALL COLOR(13,2,1)
1750 CALL COLOR(9,2,1)
1760 CALL COLOR(14,16,1)
1770 CALL CHAR(136,"1818FF3C
3C182466")
1780 CALL CHAR(144,"FFFFFFFF
FFFFFFFF")
1790 CALL CLEAR
1800 REM AUFBAU DES LABYRINT
S
1810 RESTORE 1900
1820 CALL SCREEN(6)
1830 FOR AS=1 TO 24
1840 READ A1,B1
1850 IF B1=9 THEN 1880
1860 CALL HCHAR(AS,A1,144,B1
)
1870 GOTO 1840
1880 NEXT AS
1890 GOTO 1950
1900 DATA 1,32,9,9,17,1,9,9,
5,2,8,2,13,3,19,2,26,1,30,2,
9,9,9,9,7,3,15,1,26,1,28,3,9
,9,20,2
1910 DATA 23,2,9,9,15,2,26,2
,9,9,9,9,3,3,9,3,13,3,19,2,2
6,2,9,9,22,3,9,9,2,2,7,3,11,
3,19,2
1920 DATA 26,3,9,9,9,9,5,2,8
,2,11,3,19,2,24,2,27,2,9,9,9
,9,5,2,17,3,21,3,25,2,30,2,9
,9
1930 DATA 10,2,9,9,5,2,17,3,
23,4,9,9,5,3,11,3,9,9,17,2,2
0,2,27,4,9,9,5,3,13,3,9,9,2,
2,29,2
1940 DATA 9,9,2,4,15,3,23,2,
26,2,29,2,9,9,2,2,9,9,1,32,9
,9
1950 FOR AS=1 TO 24
1960 READ A1,B1
1970 IF B1=9 THEN 2000
1980 CALL VCHAR(AS,A1,144,B1
)
1990 GOTO 1960
2000 NEXT AS
2010 CALL VCHAR(3,22,144,2)
2020 GOTO 2080
2030 DATA 9,9,1,22,11,4,24,3
,28,2,32,22,9,9,3,2,9,9,17,2
,20,1,9,9,5,2,9,3,13,2,18,3,
9,9,3,2
2040 DATA 24,3,9,9,7,4,11,3,
29,3,31,3,9,9,5,1,13,1,20,1,
22,2,9,9,17,2,9,9,9,9,5,2,15
,4,30,3
2050 DATA 9,9,17,2,22,2,24,2
,9,9,3,3,13,3,9,9,9,9,8,2,11
,2,28,3,9,9,15,3,9,9,3,3,13,
2,21,4
2060 DATA 30,3,9,9,9,2,9,9,2
3,4,25,2,27,2,9,9,11,3,9,9,9
,2,17,2,19,2,9,9,7,2,13,2,21
,2
2070 DATA 9,9,9,9,9,9
2080 RANDOMIZE
2090 FOR AS=1 TO 15
2100 X=INT(22*RND)+2
2110 Y=INT(30*RND)+2
2120 CALL GCHAR(X,Y,G)
2130 IF G=144 THEN 2100
2140 CALL HCHAR(X,Y,96)
2150 NEXT AS
2160 FOR AS=1 TO 15
2170 X=INT(15*RND)+6
2180 Y=INT(30*RND)+2
2190 CALL GCHAR(X,Y,G)
2200 IF G=144 THEN 2170
2210 IF G=96 THEN 2170
2220 CALL HCHAR(X,Y,112)
2230 NEXT AS
2240 FOR AS=1 TO 4
2250 X=INT(15*RND)+6
2260 Y=INT(30*RND)+2
2270 CALL GCHAR(X,Y,G)
2280 IF G=144 THEN 2250
2290 IF G=96 THEN 2250
2300 CALL HCHAR(X,Y,128)
2310 NEXT AS
2320 CALL HCHAR(23,4,120)
2330 CALL HCHAR(2,31,136)
2340 A=2
2350 B=31
2360 CALL KEY(0,K,S)
2370 IF S=0 THEN 2360
2380 IF K=69 THEN 2430
2390 IF K=68 THEN 2480
2400 IF K=83 THEN 2530
2410 IF K=88 THEN 2580
2420 GOTO 2360
2430 CALL GCHAR(A-1,B,G)
2440 IF G=144 THEN 2360
2450 CALL HCHAR(A,B,32)
2460 A=A-1
2470 GOTO 2620
2480 CALL GCHAR(A,B+1,G)
2490 IF G=144 THEN 2360
2500 CALL HCHAR(A,B,32)
2510 B=B+1
2520 GOTO 2620
2530 CALL GCHAR(A,B-1,G)
```

```
2540 IF G=144 THEN 2360
2550 CALL HCHAR(A,B,32)
2560 B=B-1
2570 GOTO 2620
2580 CALL GCHAR(A+1,B,G)
2590 IF G=144 THEN 2360
2600 CALL HCHAR(A,B,32)
2610 A=A+1
2620 IF G=112 THEN 2730
2630 CALL HCHAR(A,B,136)
2640 IF G=128 THEN 2750
2650 IF G=96 THEN 2680
2660 IF G=120 THEN 2710
2670 GOTO 2360
2680 PU=PU+10
2690 CALL SOUND(1000,523,5)
2700 GOTO 2360
2710 CALL SOUND(4000,131,5)
2720 GOTO 2870
2730 LE=LE+1
2740 IF LE=3 THEN 2750 ELSE
2840
2750 CALL CLEAR
2760 CALL SOUND(4000,131,5)
2770 PRINT "DU HAST DEIN LEB
EN VERWIRKT!": :
2780 PRINT "DU HAST":PU:"PUN
KTE ERREICHT": : : : :
2790 PRINT "NOCHMAL?"
2800 CALL KEY(2,K,S)
2810 IF S=0 THEN 2800
2820 IF K=2 THEN 3740
2830 END
2840 CALL HCHAR(A,B,32)
2850 CALL SOUND(2000,262,5)
2860 GOTO 2320
2870 REM
2880 CALL CHAR(120,"8181C35A
7E3C3C18")
2890 CALL CHAR(96,"183C7EDBF
FC3FFFF")
2900 CALL CHAR(42,"000028107
C1028")
2910 CALL CLEAR
2920 X=12
2930 Y=16
2940 CALL SCREEN(2)
2950 CALL COLOR(12,6,1)
2960 CALL COLOR(1,10,1)
2970 CALL COLOR(2,12,1)
2980 CALL COLOR(9,14,1)
2990 CALL COLOR(11,4,1)
3000 CALL COLOR(4,2,1)
3010 CALL CHAR(112,"81C3E77E
3C3C66C3")
3020 REM STERNENSETZUNG
3030 FOR AS=1 TO 120
3040 CALL HCHAR(INT(24*RND)+
1,INT(32*RND)+1,42,1)
3050 NEXT AS
3060 Z=INT(24*RND)+1
3070 W=INT(32*RND)+1
3080 IF INT(5*RND)=3 THEN 35
50
3090 IF INT(10*RND)=3 THEN 3
600
3100 CALL HCHAR(Z,W,112)
3110 CALL KEY(0,K,S)
3120 IF S=0 THEN 3110
3130 IF K=88 THEN 3200
3140 IF K=83 THEN 3220
3150 IF K=68 THEN 3240
3160 IF K=69 THEN 3180
3170 GOTO 3110
3180 X=X-1
3190 GOTO 3250
3200 X=X+1
3210 GOTO 3250
3220 Y=Y-1
3230 GOTO 3250
3240 Y=Y+1
3250 IF X<1 THEN 3510
3260 IF X>24 THEN 3510
3270 IF Y<1 THEN 3510
3280 IF Y>32 THEN 3510
3290 CALL GCHAR(X,Y,G)
3300 CALL HCHAR(X,Y,56)
3310 IF G=112 THEN 3420
3320 IF G=96 THEN 3570
3330 IF G=120 THEN 3620
3340 IF G=35 THEN 3450
3350 J=INT(13*RND)+1
3360 IF J=4 THEN 3380
3370 GOTO 3110
3380 Z=INT(24*RND)+1
3390 W=INT(32*RND)+1
3400 CALL HCHAR(Z,W,35)
3410 GOTO 3110
3420 CALL SOUND(2000,262,5)
3430 PU=PU+1
3440 GOTO 3060
3450 CALL CLEAR
3460 CALL SOUND(4000,523,5)
3470 PRINT "DU HAST EINE ENE
RGIESPIRALE": :"GETROFFEN!":
 :
3480 CALL COLOR(2,2,1)
3490 CALL SCREEN(8)
3500 GOTO 2780
3510 CALL CLEAR
```

```
3520 CALL SOUND(4000,523,5)
3530 PRINT "DU HAST DIE GALA
XI VERLASSEN": :
3540 GOTO 3480
3550 CALL HCHAR(Z,W,96)
3560 GOTO 3110
3570 PU=PU+5
3580 CALL SOUND(2000,440,5)
3590 GOTO 3060
3600 CALL HCHAR(Z,W,120)
3610 GOTO 3110
3620 CALL SOUND(2000,220,5)
3630 PU=PU+10
3640 GOTO 3060
3650 RESTORE 3710
3660 READ A1,B1
3670 IF B1=0 THEN 3700
3680 CALL SOUND(200*A1,B1,0)
3690 GOTO 3660
3700 RETURN
3710 DATA 3,349,1,262,3,392,
1,262,3,440,1,349,4,466,3,49
4,1,392,3,494,1,392,4,523
3720 DATA 1,587,1,659,1,698,
2,523,1,349,1,587,1,659,1,69
8,3,523
3730 DATA 1,587,1,659,1,698,
1,523,1,659,1,698,2,880,1,78
4,1,698,2,789,0,0
3740 SU=0
3750 PU=0
3760 F=0
3770 LE=0
3780 CALL COLOR(11,2,1)
3790 GOTO 710
```

VIER-KAMPF

Bei diesem Spiel können sich bis zu 10 Teilnehmer in 4 olympischen Disziplinen üben.

1. 100 METER-LAUF

Nach dem Startschuß bewegt sich der Läufer durch Drükken von beliebigen Tasten der Konsole in Richtung Ziel. Je schneller und gleichmäßiger die Tasten gedrückt werden, desto besser fällt die gelaufene Zeit aus. Zwei Fehlstarte sind erlaubt.

2. WEITSPRUNG

Die Steuerung des Anlaufes erfolgt wie unter 1 beschrieben. Absprung durch die Spacetaste. Die gesprungene Weite hängt von der Anlaufgeschwindigkeit ab. Drei Versuche sind gestattet.

3. HOCHSPRUNG

Zuerst wird die gewünschte Höhe eingegeben, dann erfolgt der Anlauf wie unter 1.

Ist der Sprung gelungen, kann man sich an einer anderen Höhe versuchen. Bei jeder Höhe sind 3 Versuche möglich, der beste wird gewertet.

4. 110 METER HÜRDENLAUF

Gelaufen wird wieder durch Drücken von beliebigen Tasten, der Sprung über die Hürden erfolgt durch die Leertaste.

Nach jeder Disziplin wird die Punktzahl angezeigt und am Ende des Wettkampfes erfolgt die Siegerehrung für die 3 besten Spieler.

Und nun wünschen wir viel Spaß beim sportlichen Computerwettkampf.

Alexander Ramthun

FEIERABENDVERKEHR

Wohnen Sie in einer Großstadt und haben schon einmal versucht, in der Rush-our eine 4-spurige Straße zu überqueren? Wenn nein, dann können Sie es mit diesem Spielprogramm ja einmal probieren. Ihre Spielfigur steht am rechten Spielfeldrand einer stark befahrenen Straße und Ihre Aufgabe ist es, die andere Straßenseite zu erreichen, ohne dabei mit einem der Autos zuammenzustoßen. Haben Sie die andere Seite dreimal lebend erreicht, so geht das Spiel in die zweite Runde, in der es die Autofahrer ein bißchen eiliger haben. Sollten Sie versuchen, mit der Spielfigur den Bildschirm zu verlassen oder kommen Sie unter die Räder, so müssen Sie dafür einen Minuspunkt in Kauf nehmen. Bei 10 Minuspunkten ist das Spiel für Sie beendet. Die Figuren, welche die andere Straßenseite schon erreicht haben, dürfen ebenfalls nicht berührt werden, da auch dies mit Minuspunkten geahndet wird.

Chr. Disch

SCHIFFVERSENKEN

Dieses Spiel kennt Ihr alle aus langweiligen Schulstunden her. Möchtet Ihr es nicht ganz gerne auch einmal zu Hause spielen und fehlt Euch der Partner dazu? Nehmt doch einfach Euren TI als Mitspieler. Er kann es genauso gut wie Ihr.

In einem 9X9 Kästchen könnt Ihr eine beliebige Anzahl von Schiffen verstekcen, der Computer macht es Euch nach. Jedes Schiff ist genau ein Kästchen groß. Wer als Erster alle Schiffe des Gegners versenkt hat, ist Sieger.

Und keine Angst, Ihr braucht nichts zuzuhalten. Der Computer schummelt nicht.

Thomas Zschake

```
100 ! ************************
110 ! *                      *
120 ! *       4-KAMPF        *
130 ! *    Copyright by      *
140 ! * Alexander Ramthun    *
150 ! *                      *
160 ! *   Bearbeitet vom     *
170 ! *       Team des       *
180 ! *  Aktuell Verlages    *
190 ! *       Muenchen       *
200 ! *                      *
210 ! * Benoetigte Geraete   *
220 ! *  TI99/4A  Konsole    *
230 ! *      Ext. Basic      *
240 ! *                      *
270 ! *  Speicherbelegung    *
280 ! *     11453 Bytes      *
290 ! ************************

295 !
300 CALL CLEAR
310 N$="4-KAMPF"
320 CALL SCREEN(3)
330 CALL MAGNIFY(2)
340 FOR I=1 TO 7 :: CALL SPR
ITE(#I,ASC(SEG$(N$,I,1)),2,3
5,20):: NEXT I
350 FOR I=1 TO 7 :: CALL MOT
ION(#I,I-3,0):: NEXT I
360 FOR I=1 TO 660 :: NEXT I
370 FOR I=1 TO 7 :: CALL MOT
ION(#I,0,0):: NEXT I
380 DISPLAY AT(2,10):"1. 100
 METER LAUF"
390 DISPLAY AT(4,10):"2. WEI
TSPRUNG"
400 DISPLAY AT(6,10):"3. HOC
HSPRUNG"
410 DISPLAY AT(8,10):"4. 110
 M HUERDEN"
420 CALL APPLAUS(3)
430 DISPLAY AT(24,5):"DRUECK
E EINE TASTE"
440 CALL KEY(0,K,S):: IF S<>
1 THEN 440
450 CALL CLEAR
460 CALL DELSPRITE(ALL)
470 DIM RANG$(10,10):: DIM P
OINTS(10):: DIM NAME$(10)
480 PLATZ=1
490 ON WARNING NEXT
500 CALL CLEAR :: DISPLAY AT
(11,6):"TEILNEHMERZAHL:" ::
ACCEPT AT(11,21)VALIDATE(DIG
IT):ANZ :: IF ANZ>10 THEN 50
0
510 CALL CLEAR :: DISPLAY AT
(1,1):"DIE SPIELERNAMEN BITT
E"
520 FOR I=1 TO ANZ
530 ACCEPT AT(I+1,1)VALIDATE
(UALPHA,"-",".")SIZE(11):NAM
E$(I):: IF NAME$(I)="" THEN
530
540 FOR LAE=1 TO 11-LEN(NAME
$(I)):: NAME$(I)=NAME$(I)&"
" :: NEXT LAE
550 FOR J=1 TO I-1
560 IF NAME$(I)=NAME$(J)THEN
 530
570 NEXT J
580 NEXT I
590 CALL CLEAR :: DISPLAY AT
(12,10):"DURCHGAENGE:" :: AC
CEPT AT(12,22)VALIDATE(DIGIT
):DURCHG
600 FOR DURCH=1 TO DURCHG
610 FOR EDURCH=1 TO ANZ
620 CALL CLEAR
630 HUERDE=0
640 DISPLAY AT(12,12):NAME$(
EDURCH)
650 IF EDURCH=1 AND DURCH=1
THEN 700
660 IF POINTS(EDURCH)=BPOINT
S THEN 700
670 DISPLAY AT(15,3):"SIE BR
AUCHEN";BPOINTS-POINTS(EDURC
H)+1
680 DISPLAY AT(16,1):"PUNKTE
 UM IN FUEHRUNG ZU" :: DISPL
AY AT(17,7):"GEHEN"
690 FOR I=1 TO 1000 :: NEXT
I :: GOTO 710
700 FOR I=1 TO 500 :: NEXT I
710 CALL CLEAR
720 CALL CHAR(81,"3838103E58
186402",64,"1C1D0AFC18285088
",62,"2112945B3F0B3000",61,"
3838103E303F3807")
730 GOSUB 750
740 GOTO 850
750 CALL CHAR(130,"FF",120,"
7E81A5818199423C",128,"01020
40810204080",124,RPT$("10",8
),126,"00000000000000FF")
760 CALL HCHAR(2,1,120,192)
770 J=17
780 CALL COLOR(9,9,9,13,2,9)
790 CALL HCHAR(8,1,126,32)
800 DISPLAY AT(9,1):"ö  ö  ö
  ö  ö  ö  ö  ö  ö  ö"
```

```
810 CALL HCHAR(10,1,97,384)
820 CALL HCHAR(10,1,130,64):
: CALL HCHAR(13,1,130,32)::
CALL HCHAR(15,1,130,32):: CA
LL HCHAR(18,1,130,32):: CALL
 HCHAR(21,1,130,32)
830 FOR I=10 TO 20 :: CALL H
CHAR(I,J,128):: J=J-1 :: NEX
T I
840 RETURN
850 A=73 :: F=0
860 CALL SPRITE(#1,81,2,109,
73)
870 CALL APPLAUS(2)
880 GOSUB 900
890 IF F=3 THEN 1040 ELSE 10
30
900 DISPLAY AT(22,2):"DRUECK
E EINE TASTE"
910 CALL KEY(0,K,S):: IF S<>
1 THEN 910
920 DISPLAY AT(22,2):""
930 CALL SOUND(-200,600,0)::
 DISPLAY AT(1,12):"ACHTUNG"
940 FOR I=1 TO 50
950 CALL KEY(0,K,S):: IF S=1
 THEN 1080
960 NEXT I
970 CALL SOUND(-200,600,0)::
 DISPLAY AT(1,12):"FERTIG"
980 FOR I=1 TO 50
990 CALL KEY(0,K,S):: IF S<>
0 THEN 1080
1000 NEXT I
1010 CALL SOUND(150,-7,0)
1020 RETURN
1030 T=T+.1 :: CALL KEY(0,K,
S):: IF S<>1 THEN 1030 :: A=
A+3 :: IF A>=250 THEN 1070 :
: IF A=73 THEN 1040 :: CALL
LOCATE(#1,109,A):: CALL SOUN
D(-10,-5,0):: GOTO 1030
1040 DISPLAY AT(1,5):USING "
##.##":T :: IF T=0 THEN 1050
 ELSE CALL APPLAUS(INT(20/T)
)
1050 IF T=0 THEN P=0 :: GOTO
 1110 ELSE P=INT(1000/T*10):
: IF P<0 THEN P=0
1060 GOTO 1110
1070 A=1 :: GOTO 1030
1080 CALL SOUND(200,110,0)
1090 F=F+1 :: DISPLAY AT(23,
3):F;".FEHLSTART" :: FOR I=1
 TO 400 :: NEXT I :: IF F=3
THEN RETURN ELSE 900
1100 P=0
1110 DISPLAY AT(24,2):"PUNKT
E:";P
1120 FOR I=1 TO 500 :: NEXT
I
1130 CALL DELSPRITE(#1):: CA
LL CLEAR
1140 DISPLAY AT(24,2):"PUNKT
E:";P
1150 T=0
1160 CALL COLOR(13,9,15,8,15
,15,1,15,1)
1170 CALL CHAR(128,"FFFEFCF8
F0E0C08",35,"FFFEFCF8F0E0C08
")
1180 CALL HCHAR(2,1,120,192)
:: CALL HCHAR(8,1,126,32)::
DISPLAY AT(9,1):"ö  ö  ö  ö
 ö  ö  ö  ö  ö  ö"
1190 FOR I=15 TO 20 :: CALL
HCHAR(I,1,97,35-I):: DISPLAY
 AT(I,34-I):CHR$(128):: NEXT
 I
1200 CALL HCHAR(15,22,88,11)
:: CALL HCHAR(16,21,88,12)::
 CALL HCHAR(17,20,88,12):: C
ALL HCHAR(18,19,88,12):: CAL
L HCHAR(19,18,88,12)
1210 CALL HCHAR(20,17,88,12)
1220 CALL HCHAR(17,32,35)::
CALL HCHAR(18,31,35):: CALL
HCHAR(19,30,35):: CALL HCHAR
(20,29,35)
1230 CALL SPRITE(#1,81,2,125
,2)
1240 CALL APPLAUS(2)
1250 FOR D=1 TO 3
1260 A=2 :: CALL SPRITE(#1,8
1,2,125,A)
1270 CALL KEY(0,K,S):: IF S<
>1 THEN 1270
1280 T=T+1 :: CALL KEY(0,K,S
):: IF S<>1 THEN 1280 :: IF
K=32 THEN 1290 :: A=A+2 :: C
ALL LOCATE(#1,125,A):: IF A>
126 THEN 1370 :: CALL SOUND(
-10,-5,0):: GOTO 1280
1290 T=T+126-A :: M=142-T ::
 IF M<0 THEN M=0
1300 CALL PATTERN(#1,61)
1310 CALL MOTION(#1,-2,INT(M
/13)):: FOR I=1 TO M/1.5 ::
CALL SOUND(-100,600+I*10,8):
: NEXT I
```

```
1320 CALL MOTION(#1,2,INT(M/
13)):: FOR I=M/1.5 TO 1 STEP
 -1 :: CALL SOUND(-100,600+I
*10,8):: NEXT I
1330 CALL MOTION(#1,0,0):: C
ALL POSITION(#1,X;Y):: Y=Y-1
26 :: WEITE(D)=Y/10-2 :: IF
WEITE(D)<0 THEN WEITE(D)=0
1340 DISPLAY AT(20+D,3):D;".
VERSUCH:";WEITE(D);"METER" :
: CALL APPLAUS(INT(WEITE(D)/
3))
1350 IF WEITE(D)>HW THEN HW=
WEITE(D):: GOTO 1390
1360 GOTO 1390
1370 WEITE(D)=0
1380 GOTO 1340
1390 A,T,M=0 :: NEXT D
1400 FOR I=1 TO 500 :: NEXT
I
1410 FOR I=21 TO 23 :: DISPL
AY AT(I,1):"" :: NEXT I
1420 DISPLAY AT(22,1):"HOECH
STE WEITE:";HW;"METER" :: CA
LL APPLAUS(INT(HW/3))
1430 P=P+HW*100
1440 DISPLAY AT(24,2):"PUNKT
E:";P
1450 FOR I=1 TO 400 :: NEXT
I :: CALL DELSPRITE(#1):: CA
LL CLEAR
1460 CALL COLOR(13,2,9)
1470 A,B,T=0
1480 CALL HCHAR(2,1,120,192)
:: CALL HCHAR(8,1,126,32)::
DISPLAY AT(9,1):"ö..ö..ö..ö.
.ö..ö..ö..ö..ö.."
1490 DISPLAY AT(24,2):"PUNKT
E:";P
1500 CALL HCHAR(10,1,97,417)
1510 CALL COLOR(11,15,9,10,2
,15)
1520 CALL CHAR(112,"FF7F3F1F
0F070301",113,"80C0E0F0F8FCF
EFF",104,RPT$("10",8),128,RP
T$("10",8),131,"804020100804
0201")
1530 J=21 :: FOR I=16 TO 20
:: CALL HCHAR(I,J,112):: CAL
L HCHAR(I,J+1,88,6):: CALL H
CHAR(I,J+7,113):: J=J+1 :: N
EXT I
1540 CALL VCHAR(10,20,128,7)
:: CALL HCHAR(15,25,128):: C
ALL VCHAR(16,25,104,4):: CAL
L VCHAR(20,25,128,2)
1550 DISPLAY AT(23,2):"HOEHE
?" :: ACCEPT AT(23,10)VALIDA
TE(DIGIT,".")SIZE(4):H
1560 IF H>5 THEN CALL SOUND(
300,110,0):: GOTO 1550
1570 T,A,B=0
1580 IF H<=HH THEN CALL SOUN
D(300,110,0):: GOTO 1550
1590 H=H*17
1600 FOR V=1 TO 3
1610 DISPLAY AT(1,1):V;".VER
SUCH UEBER";H/17;"M"
1620 CALL SPRITE(#2,131,2,12
8-H,160):: CALL SPRITE(#3,13
1,2,145-H,177)
1630 A=40 :: B=79 :: CALL SP
RITE(#1,81,2,B,A)
1640 CALL KEY(0,K,S):: IF S<
>1 THEN 1640
1650 CALL KEY(0,K,S):: IF S<
>1 THEN 1660 :: T=T+1 :: A=A
+2 :: CALL LOCATE(#1,B,A)::
CALL SOUND(-10,-5,0):: IF A=
94 THEN 1670 :: GOTO 1650
1660 T=T-1.5 :: GOTO 1650
1670 CALL KEY(0,K,S):: IF K=
32 THEN 1690 :: IF S<>1 THEN
 1680 :: T=T+1 :: A=A+2 :: B
=B+2 :: CALL LOCATE(#1,B,A):
: CALL SOUND(-10,-5,0):: IF
A>166 THEN 1870 :: GOTO 1670
1680 T=T-1.5 :: GOTO 1670
1690 IF A<130 THEN 1670
1700 CALL POSITION(#1,AH,U)
1710 SK=T*1.2
1720 CALL PATTERN(#1,64)
1730 FOR I=1 TO SK/1.5 :: CA
LL MOTION(#1,(-SK+I)/5,I/10)
:: NEXT I
1740 CALL PATTERN(#1,62)
1750 CALL MOTION(#1,0,0):: C
ALL POSITION(#1,X,Y):: IF X+
14>AH-H THEN 1810
1760 FOR I=1 TO SK/1.7 :: CA
LL MOTION(#1,INT(SK+I)/8,7):
: NEXT I
1770 CALL MOTION(#1,0,0)
1780 DISPLAY AT(1,1):H/17;"M
 UBERSPRUNGEN" :: CALL APPLA
US(INT(H/17))
1790 HH=H/17 :: HHP=H/17*200
1800 T=0 :: GOTO 1550
1810 IF X>AH-H THEN 1820 ELS
E 1840
```

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE

KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre TI REVUE nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Euer" Kiosk nicht beliefert wurde? Kein Problem! Für ganze 30 DM liefern wir Euch per Post sechs Hefte ins Haus (Ausland 40 DM). Einfach den Bestellschein auf der nächsten Seite ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). TI REVUE kommt dann pünktlich ins Haus.

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen– Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für sechs Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

DAS SUPER-SONDER-ANGEBOT: PRIVATE KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen ab sofort die TI-Revue: KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!

Wir veröffentlichen nur Kleinanzeigen privater Inserenten, keine gewerblichen Anzeigen. Die kosten pro Millimeter DM 2,50 plus Mehrwertsteuer!

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!

Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!

Die Insertion ist nicht vom Kauf des Heftes abhängig!

Wir behalten uns vor, Anzeigen, die nicht zum Themenkreis des Heftes – Computer – gehören, nicht abzudrucken oder sie nur insoweit zu berücksichtigen, wie es der Umfang des kostenlosen Anzeitenteils zuläßt.

KASSETTEN

Selbstverständlich gibt es in diesem Heft wieder einen Kassetten-Service. Dabei können wir Ihnen allerdings die Qual der Wahl nicht ersparen. Denn: In Ihrem Interesse haben wir dieses Special sozusagen „geviertelt".

Viertel 1: Alle Utilities und Anwender-Programme in TI Basic
Viertel 2: Alle Spiele in TI Basic
Viertel 3: Alle Utilities/Anwenderprogramme in Extended Basic
Viertel 4: Alle Spiele in Extended Basic.

Und weil es uns Spaß macht, haben wir noch ein fünftes Viertel angehängt: den Disketten-Service. Da sind dann allerdings alle Programme drauf (abgesehen von zwei Ausnahmen). Und nun zu den harten, nackten Geschäftsbedingungen.

Sie können zahlen
Gegen Nachnahme plus Postgebühren
Gegen Bankabbuchung am Versandtag
Gegen Vorausrechnung, wobei am Tag des Geldeinganges der Versand erfolgt.

Sie können kombinieren:
Jedwedes Programmpaket miteinander — etwa 1 + 4 oder 1,2,3 oder nur 4 und so weiter.

ANZEIGENSERVICE

Die große Börse für jeden Zweck in der TI REVUE. Kostenlos für Privat-Inserenten. Spott—billig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Achtung! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt nicht TI-typisch ist oder gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatanbieter: maximal acht Zeilen à 28 Anschläge.

Vorname ____________

(PLZ) Ort ____________

TI REVUE
Anzeigenabteilg.
Postfach 1107
8044 Lohhof

ABO SERVICE-KARTE

6
TI

Coupon

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten sechs Ausgaben an untenstehende Anschrift.

Name ____________
Vorname ____________
Straße/Hausnr. ____________
Plz/Ort ____________
Ich bezahle:
☐ per beiliegendem Verrechnungsscheck
☐ gegen Rechnung
☐ bargeldlos per Bankeinzug von meinem Konto
bei (Bank) und Ort ____________
Kontonummer ____________
Bankleitzahl ____________
(steht auf jedem Kontoauszug)
Unterschrift ____________

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift ____________

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

TI REVUE
Abo-Service 6
Postfach 1107
8044 UNTERSCHLEISSHEIM

SERVICE

Das sind die Preise: Pro einzelnes Programm-Paket DM 20,–
Für zwei Programmpakete DM 35,– (5,– DM Ersparnis)
Für drei Programmpakete DM 50,– (10,– DM Ersparnis)
Für alle vier Programmpakete DM 60,– (20,– DM Ersparnis)
(Jedes Programmpaket auf Kassette!)
Für die komplette Diskette – alle vier Programm-Pakete – DM 75,– (hier ist keine Zusammenstellung möglich!)

Die Inhalte der einzelnen Programm-Pakete ersehen Sie aus nachfolgender Übersicht, die Komplett-Diskette enthält aller vier Programm-Pakete!

ACHTUNG: KEIN GELD EINSENDEN! NUR DEN BERECHTIGUNGSCOUPON! WIR MELDEN UNS, WENN DIE WARE VERSANDBEREIT IST!!! SELBSTVERSTÄNDLICH VOLLES UMTAUSCHRECHT BEI NICHTFUNKTIONIEREN.

Programm-Paket 1: TI BASIC!
Funktionsplot, Determinante, Quadr. Ergänzungen, Französische Zahlen, Dateiverwaltung, Tabellenerstellung, Jahreszinsrechnung (zusammen ca. 28 Kb)

Programm-Paket 2: TI BASIC!
Blumengießen, Black Hole, Schiffe versenken, Pushit, Codo, Minigolf, Hammurabi (zusammen ca. 61 Kb)

Programm-Paket 3: EXT. BASIC!
Betriebssystem, Sprachschatz, Pixelmaster, Soundgenerator, Sternenhimmel, Videothek, Texteditor, Kontokorrent (zusammen ca. 61 Kb)

Programm-Paket 4: EXT. BASIC!
Spinnenspiel, Heinrich, der Schreckensherrscher, Vulkanien, Feierabend, Vierkampf, Break out, Monopoly, Wortsuchspiel. (zusammen ca. 69 Kb)

ACHTUNG:!': Wir weisen ausdrücklich darauf hin, daß die Programme „PHYSIK" und „BROKER" **nicht** in diesem Angebot enthalten sind! Sollten Sie daran Interesse haben, wenden Sie sich bitte direkt an den jeweiligen Autor, dessen Adresse in diesen Programmen enthalten ist! Rückfragen beim Kassettenservice sind zwecklos! Er hat auch keinen Einfluß auf die von den Autoren geforderten Preise!

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen
Programmsammlung 1 ( ) Programmsammlung 2 ( )
Programmsammlung 3 ( ) Programmsammlung 4 ( )
die Komplett-Diskette ( )
Zutreffendes bitte ankreuzen!
Ich zahle:
Gegen Nachnahme plus Postgebühren ( )
Gegen Vorausrechnung ( ) – Versand am Tag des Geldeinganges
Gegen Bankabbuchung am Versandtag ( )
Meine Bank (mit Ortsname) .....
Meine Kontonummer .....
Meine Bankleitzahl ..... (steht auf jedem Bankauszug)
Vorname ..... Nachname .....
Str./Nr. ..... PLZ / Ort .....
Hiermit bestätigen ich mit meiner Unterschrift, Ihre Verkaufsbedingungen gelesen zu haben und zu akzeptieren.

Unterschrift .....

Bitte ausschneiden und einsenden an
TI-SPECIAL
KASSETTENSERVICE
Postfach 1107
8044 Unterschleißheim

COMPUTER AKTUELL

Nr. 5/85 Spezial-Ausgabe - B 20131 F

100 Seiten KAUFBERATUNG

DM 9,80/ÖS 80/SFR 9,80

COMPUTER 85: BESSER SCHNELLER BILLIGER!

NEU!

HANNOVER-MESSE: ALLE NEUHEITEN!

COMPUTERN LEICHT GEMACHT

KAMPF DER GIGANTEN:
ATARI CONTRA COMMODORE

IM TEST:
YASHICA MSX
PHILIPS MSX
SPECTRA MSX
CETEC MSX
SONY MSX

COMMODORE PLUS 4
SPECTRUM PLUS

DER HOME COMPUTER IST TOT
ES LEBE DER PRIVATE PC

◇◇◇◇◇◇

JETZT AN IHREM KIOSK

◇◇◇◇◇◇

WAS IST DRAN AN MSX?

GEWINNEN SIE EINEN COMPUTER

SERVICE
TIPS & TRICKS

```
1820 FOR I=1 TO SK/1.7 :: CA
LL MOTION(#1,INT(SK+I)/8,7):
: NEXT I
1830 CALL MOTION(#1,0,0):: F
OR I=1 TO 500 :: NEXT I :: G
OTO 1890
1840 CALL MOTION(#1,10,0,#2,
10,0,#3,10,0)
1850 CALL POSITION(#1,Z,X)::
 IF Z>=140 THEN 1860 :: GOTO
 1850
1860 CALL MOTION(#2,0,0,#3,0
,0,#1,0,0):: GOTO 1880
1870 DISPLAY AT(1,1):"FRUEHE
R ABSPRINGEN"
1880 FOR I=1 TO 500 :: NEXT
I
1890 CALL DELSPRITE(ALL):: T
,A,B=0
1900 NEXT V
1910 DISPLAY AT(1,1):"UEBERS
PRUNGEN:";HH;"METER"
1920 P=P+HHP*2 :: DISPLAY AT
(24,2):"PUNKTE:";P :: CALL A
PPLAUS(INT(HH))
1930 FOR I=1 TO 600 :: NEXT
I
1940 HUERDE=1 :: CALL CLEAR
:: CALL COLOR(12,2,1):: GOSU
B 750
1950 CALL CHAR(140,"040C1424
40404040")
1960 CALL SPRITE(#2,140,2,11
6,37,#3,140,2,116,150,#4,140
,2,116,230)
1970 A=73 :: CALL SPRITE(#1,
81,2,110,A)
1980 DISPLAY AT(24,2):"PUNKT
E:";P :: CALL APPLAUS(2)
1990 F=0
2000 GOSUB 900
2010 IF F=3 THEN 2160
2020 T=T+.1 :: CALL KEY(0,K,
S):: IF K=32 THEN 2060 :: IF
 S<>1 THEN 2020 :: A=A+3 ::
IF A>250 THEN 2040 :: CALL L
OCATE(#1,110,A):: CALL SOUND
(-10,-5,0):: IF A<74 AND A>7
0 THEN 2160 :: CALL COINC(AL
L,L)
2030 IF L=-1 THEN 2050 :: GO
TO 2020
2040 A=1 :: GOTO 2020
2050 T=T+5 :: CALL SOUND(-10
0,110,0):: GOTO 2020
2060 CALL MOTION(#1,-7,8)
2070 FOR I=1 TO 12
2080 CALL POSITION(#1,X,Y)::
 IF Y=72 THEN 2150
2090 NEXT I
2100 CALL MOTION(#1,7,8)
2110 FOR I=1 TO 12
2120 CALL POSITION(#1,X,Y)::
 IF Y=72 THEN 2150
2130 NEXT I
2140 CALL MOTION(#1,0,0):: T
=T+2 :: A=Y :: GOTO 2020
2150 CALL MOTION(#1,0,0)
2160 DISPLAY AT(1,1):USING "
###.##":T :: IF T=0 THEN 217
0 ELSE P=P+INT(1000/T*10)
2170 DISPLAY AT(24,2):"PUNKT
E:";P :: IF T=0 THEN 2180 ::
 CALL APPLAUS(INT(15-T))
2180 FOR I=1 TO 500
2190 NEXT I
2200 CALL DELSPRITE(ALL):: T
,A,B,HW,W,H,HH,HHP=0
2210 IF P>WR THEN 2220 ELSE
2230
2220 CALL REKORD(P,WR):: WR=
P
2230 POINTS(EDURCH)=POINTS(E
DURCH)+P
2240 FOR PRI=1 TO ANZ
2250 FOR PRIN=1 TO ANZ :: IF
 NAME$(PRI)=NAME$(PRIN)THEN
2270
2260 IF POINTS(PRI)<POINTS(P
RIN)THEN PLATZ=PLATZ+1
2270 NEXT PRIN
2280 IF PLATZ=1 THEN BPOINTS
=POINTS(PRI)
2290 IF RANG$(PLATZ,1)<>"" T
HEN 2300 ELSE 2330
2300 FOR I=2 TO ANZ :: IF RA
NG$(PLATZ,I)<>"" THEN 2320
2310 RANG$(PLATZ,I)="  "&NAM
E$(PRI)&" "&STR$(POINTS(PRI)
):: GOTO 2340
2320 NEXT I :: GOTO 2340
2330 RANG$(PLATZ,1)=STR$(PLA
TZ)&"."&NAME$(PRI)&" "&STR$(
POINTS(PRI))
2340 PLATZ=1 :: NEXT PRI
2350 CALL CLEAR :: CALL CHAR
SET
2360 FOR I=1 TO ANZ
2370 IF RANG$(I,1)="" THEN 2
420
```

```
2380 ZEI=ZEI+1 :: DISPLAY AT
(ZEI,1):RANG$(I,1)
2390 FOR J=2 TO ANZ :: IF RA
NG$(I,J)<>"" THEN 2400 ELSE
2420
2400 ZEI=ZEI+1 :: DISPLAY AT
(ZEI,1):RANG$(I,J)
2410 NEXT J
2420 NEXT I
2430 DISPLAY AT(24,1):"DRUEC
KE EINE TASTE"
2440 P,ZEI=0
2450 CALL KEY(0,K,S):: IF S=
0 THEN 2450
2460 CALL CLEAR :: DISPLAY A
T(12,7):"BITTE WARTEN"
2470 IF DURCHG=DURCH AND ANZ
=EDURCH THEN 2570
2480 Z=1
2490 FOR I=1 TO 10
2500 FOR J=1 TO 10
2510 IF RANG$(I,J)="" THEN 2
540
2520 RANG$(I,J)=""
2530 NEXT J
2540 NEXT I
2550 NEXT EDURCH
2560 NEXT DURCH :: GOTO 2580
2570 CALL SIEGER(RANG$(1,1),
RANG$(2,1),RANG$(3,1))
2580 DISPLAY AT(24,1):"NOCH
EIN SPIEL(J/N)?"
2590 CALL KEY(0,K,S):: IF S=
0 THEN 2590
2600 IF K=74 OR K=106 THEN 2
620
2610 END
2620 CALL CLEAR :: CALL DELS
PRITE(ALL):: CALL CHARSET ::
 FOR I=1 TO 12 :: CALL COLOR
(I,2,1):: NEXT I :: RUN
2630 SUB APPLAUS(ST)
2640 FOR I=1 TO ST
2650 CALL SOUND(1000,-7,0)
2660 FOR J=1 TO INT(RND*3)+1
 :: CALL SOUND(600,-7,3,200,
0):: CALL SOUND(200,-7,0)
2670 CALL KEY(0,K,S):: IF S<
>0 THEN SUBEXIT
2680 NEXT J :: NEXT I
2690 SUBEND
2700 SUB REKORD(NW,W)
2710 CALL CLEAR :: CALL CHAR
SET
2720 DISPLAY AT(1,1):"BRAVO"
2730 DISPLAY AT(3,1):"SIE HA
BEN DEN REKORD UM"
2740 DISPLAY AT(5,1):NW-W;"P
UNKTE UEBERTROFFEN"
2750 DISPLAY AT(7,1):"REKORD
:";NW
2760 DISPLAY AT(24,1):"DRUEC
KE EINE TASTE"
2770 CALL KEY(0,K,S):: IF S<
>1 THEN 2770 :: CALL CLEAR :
: DISPLAY AT(10,12):"BITTE W
ARTEN"
2780 SUBEND
2790 SUB SIEGER(P1$,P2$,P3$)
2800 IF P2$="" THEN P2$="2.C
OMPUTER"
2810 IF P3$="" THEN P3$="3.C
OMPUTER"
2820 CALL CLEAR :: CALL MAGN
IFY(2):: CALL SCREEN(3)
2830 CALL CHAR(119,"7E81A581
8199443C",126,"0000000000000
OFF",124,RPT$("10",8),96,"0"
)
2840 CALL HCHAR(1,1,119,192)
:: CALL HCHAR(8,1,126,32)::
DISPLAY AT(9,1):"ö..ö..ö..ö.
.ö..ö..ö..ö..ö.."
2850 CALL COLOR(1,2,7,2,2,7,
3,2,7,4,2,7,5,2,7,6,2,7,7,2,
7,8,2,7,9,2,7,12,2,7)
2860 CALL HCHAR(10,1,96,320)
2870 CALL CHAR(97,"00FFFFFFF
F",98,"FFFFFFFFFF",99,"0000F
FFFFF",107,"1CFFFFFFFF",108,
"1C1C081C1C1C1414",109,"14FF
FFFFFF")
2880 CALL CHAR(103,"3838107C
7BB82442",104,"3838103838381
030",105,"1C1C081C1C1C1010",
106,"1C1C081C1C1C0404")
2890 CALL SPRITE(#1,97,2,140
,120,#2,98,2,140,136,#3,99,2
,140,152)
2900 CALL SPRITE(#5,103,2,64
,217,0,-5,#6,104,2,64,230,0,
-5,#4,103,2,64,243,0,-5)
2910 CALL POSITION(#5,A,B)::
 IF B<=122 THEN 2960
2920 CALL PATTERN(#4,104,#5,
103,#6,104)
2930 FOR J=1 TO 6 :: NEXT J
2940 CALL PATTERN(#4,103,#5,
104,#6,103)
2950 FOR J=1 TO 6 :: NEXT J
:: GOTO 2910
2960 CALL MOTION(#4,5,0,#5,5
,0,#6,5,0)
```

```
2970 CALL POSITION(#5,A,B)::
 IF A>=127 THEN 3020
2980 CALL PATTERN(#4,105,#5,
106,#6,105)
2990 FOR J=1 TO 6 :: NEXT J
3000 CALL PATTERN(#4,106,#5,
105,#6,106)
3010 FOR J=1 TO 6 :: NEXT J
:: GOTO 2970
3020 CALL MOTION(#4,0,0,#5,0
,0,#6,0,0)
3030 CALL PATTERN(#1,107,#4,
108,#5,108,#6,108)
3040 FOR I=1 TO 200 :: NEXT
I
3050 DISPLAY AT(10,1):P1$ ::
 CALL LOCATE(#6,124,136):: C
ALL APPLAUS(4)
3060 FOR I=1 TO 200 :: NEXT
I
3070 CALL PATTERN(#1,109)
3080 DISPLAY AT(11,1):P2$ ::
 CALL LOCATE(#5,126,120):: C
ALL APPLAUS(3)
3090 FOR I=1 TO 200 :: NEXT
I
3100 DISPLAY AT(12,1):P3$ ::
 CALL LOCATE(#4,128,152):: C
ALL APPLAUS(3)
3110 DISPLAY AT(24,7):"DRUEC
KE EINE TASTE"
3120 CALL KEY(0,K,S):: IF S<
>1 THEN 3120
3130 SUBEND
```

EINKAUFSGEMEINSCHAFTEN: WAS IST DAS UND WAS NÜTZEN SIE?

Ich war sicherlich nicht der erste Computerbenutzer der einmal eine Kleinanzeige „Bilde Einkaufsgemeinschaft zwecks günstigeren Preisen. Info bitte bei... etc. ...“ gelesen hat und sich dabei überlegte, ob man dem Inserenten schreiben sollte.
Tatsächlich hatte und habe ich als Schüler nicht gerade übermäßig viel Geld und mein Hobby ist eben nicht das billigste. Mal braucht man ein paar neue Disketten, mal einen neuen Joystick, einen Drucker hätte man auch gerne, nicht zu vergessen, die 80-Zeichen-Karte und und und . . .
Also kam mir dieses Angebot, als Mitglied einer Einkaufsgemeinschaft billiger als sonst einkaufen zu können, sehr gelegen. (Vielleicht konnte man ja sogar einige Sachen mit kleinen Gewinn weiterverkaufen?!?)
Dennoch überlegte ich mehrfach, ob ich schreiben sollte, denn sicherlich gab und gibt es viele schwarze Schafe unter den Kleininseratenaufgeber.
Und wer gibt schon Geld für eine Kleinanzeige aus, wenn er nicht denkt, daß es sich für ihn lohnt?
(Ganz abgesehen davon, daß eine Einkaufsgemeinschaft für den Leiter derselben sicherlich ziemlich viel Arbeit mit sich bringt.)
Doch trotz dieser Überlegungen hatte ich um ein Info geschrieben, denn die 80 Pfennig + 1 Blatt Papier + 1 Briefumschlag + 1 Freiumschlag konnte ich mir noch leisten.
Nachdem ich mein Schreiben abgeschickt hatte und mittlerweile viele Monde über das Land gezogen waren, ohne daß ich ein Info erhalten hatte, raffte ich mich auf und schrieb um ein neues Info.
„Das fängt ja gut an“, dachte ich, als ich den zweiten Brief abschickte.
Erst nach dem zweiten Versuch kam ein kleiner Brief ins Haus geflattert (allerdings erst nach über 2 Wochen).
Die Hoffnung hatte ich eigentlich schon aufgegeben, und so war die Freude um so größer.
Im Brief konnte ich dann lesen, daß der Andrang groß gewesen sei, und die Beantwortung meiner Anfragen!! daher etwas lange gedauert hätte.
Weiterhin lobte man mich einen fortschrittlichen Menschen, der eine noch fortschrittlichere Idee unterstützen wollte und verwies mich dann mit freundlichen Grüßen auf den beiliegenden Angebotszettel.
Hatte ich bisher noch alles bestens überstanden, so konnte ich mich beim Anblick der Preise doch eines kleinen Schocks nicht erwehren.
So viele Zuschriften konnten wohl doch nicht eingegangen sein, denn die Preise, die mir da geboten wurden, waren auf den ersten Blick zwar recht günstig, doch leider ohne Mehrwertsteuer und die mußte man eben mitberechnen, so daß ich bald bemerkte, daß ich dieselben Produkte im Laden nebenan zum selben Preis bekommen konnte.
Dabei kann ich mir gut vorstellen, daß eine Einkaufsgemeinschaft bei größeren Bestellungen sicherlich bessere Preise gemacht bekommt, als der einzelne Verbraucher.
Doch bei der „Einkaufsgemeinschaft“ des Kleininserenten war dieser günstigere Preis anscheinend zu Gunsten des Leiters erhöht worden.
Es ging nicht um niedrige Preise sondern um einen guten Gewinn.
Und solche Gewinnsucht dürfte so ziemlich jede Einkaufsgemeinschaft zerstören können, denn die Vorteile einer ECHTEN Einkaufsgemeinschaft liegt klar auf der Hand:

Bitte lesen Sie weiter auf Seite 85

```
100 ! ************************
110 ! *                      *
120 ! * FEIERABENDVERKEHR    *
130 ! *    Copyright by      *
140 ! *  Christian  Disch    *
150 ! *                      *
160 ! *   Bearbeitet vom     *
170 ! *      Team des        *
180 ! *  Aktuell Verlages    *
190 ! *      Muenchen        *
200 ! *                      *
210 ! * Benoetigte Geraete   *
220 ! *  TI99/4A  Konsole    *
230 ! *     Ext. Basic       *
240 ! *    Joystick (1)      *
250 ! *                      *
270 ! *  Speicherbelegung    *
280 ! *     4106 Bytes       *
290 ! ************************
295 !
300 !   *** TITELBILD ***
310 !
320 CALL SCREEN(13):: CALL C
HAR(64,"3C4299A1A199423C")
330 DISPLAY AT(6,9)ERASE ALL
:"VERSUCHEN SIE"
340 DISPLAY AT(9,5):"BEI FEI
ERABENDVERKEHR"
350 DISPLAY AT(12,2):"DIE ST
RASSE ZU UEBERQUEREN"
360 DISPLAY AT(17,4):"© 1984
 CHRISTIAN DISCH"
370 ANZAHL=25 :: GOSUB 1060
:: RESTORE 1140
380 DISPLAY AT(23,2):"BITTE
ROTE TASTE DRUECKEN"
390 CALL KEY(1,TASTE,ST)
400 IF TASTE<>18 THEN 390 EL
SE CALL CLEAR
410 CALL COLOR(1,15,15)
420 CALL MAGNIFY(3)
430 TEMPO=4
440 ZAHL=28 :: ANZAHL=11
450 CALL CHAR(118,RPT$("F",1
6))
460 CALL COLOR(11,13,1):: CA
LL VCHAR(1,1,118,24):: CALL
VCHAR(1,2,118,24):: CALL VCH
AR(1,32,118,24)!RAND
470 !* ZEICHENDEFINITIONEN *
480 !
490 CALL CHAR(96,"0003030707
0F7FFFFFFFFFFFFF3C180000FEFE
FEFEFEFEFEFEFEFEFEFEFE3C18")
500 CALL CHAR(100,"000000000
000FFFFFFFFFFFFFF3C18000000F
0F8F8FCFCFEFEFEFEFEFE3C18")
510 CALL CHAR(104,"002020202
3272F2F7FFFFFFFFF3C180000000
000F0F8FCFCFEFEFEFEFE3C18")
520 CALL CHAR(108,"010101010
10F1F3F7FFFFFFFFF3C180000000
0000080C0E0F0FEFEFEFE3C18")
530 CALL CHAR(112,"83C7E7673
31F1F0F0F0F0F0E04043C7CC1E3E
7E6CCF8F8F0F0F0F07020203C3E"
,120,"0000FFFFFFFF")
540 CALL COLOR(12,9,15):: FO
R I=2 TO 9 :: CALL COLOR(I,2
,15):: NEXT I
550 !
560 ! *** SPIELAUFBAU ***
570 !
580 DISPLAY AT(1,16):"1. DUR
CHGANG" :: DISPLAY AT(24,1):
"ZEIT:.............FEHLER: 0
" :: CALL HCHAR(24,8,120,10)
590 !
600 ! *** AUTOS SETZEN ***
610 !
620 RANDOMIZE :: A=INT(256*R
ND)+1 :: B=INT(256*RND)+8 ::
 C=INT(256*RND)+16
630 IF A>235 OR B>235 OR C>2
35 THEN 620
640 IF ABS(A-B)<=33 OR ABS(A
-C)<=33 THEN 620
650 IF ABS(B-C)<=33 THEN 620
660 CALL SPRITE(#1,96,7,74,A
,0,-TEMPO,#2,96,7,74,B,0,-TE
MPO,#3,100,3,102,A,0,TEMPO+5
,#4,100,3,102,B,0,TEMPO+5)
670 CALL SPRITE(#5,104,5,130
,A,0,-TEMPO-10,#6,104,5,130,
B,0,-TEMPO-10,#7,104,5,130,C
,0,-TEMPO-10)
680 CALL SPRITE(#8,108,11,46
,A,0,TEMPO+12,#9,108,11,46,B
,0,TEMPO+12,#10,108,11,46,C,
0,TEMPO+12)
690 !
700 ! *** SPIELABLAUF ***
710 !
720 CALL SPRITE(#28,112,14,1
61,125)
730 CALL JOYST(1,X,Y)
740 CALL KEY(1,TASTE,ST)
750 IF TASTE=5 THEN 1200
760 ZEIT=ZEIT+1
770 IF ZEIT/20=INT(ZEIT/20)T
HEN CALL HCHAR(24,ZEIT/20+7,
32):: IF ZEIT>=200 THEN GOSU
B 920 :: ZEIT=0 :: CALL HCHA
R(24,8,120,10):: GOTO 720
```

```
780 CALL MOTION(#28,-Y/4*11,
X/4*11)
790 CALL COINC(ALL,TREFFEN):
: IF TREFFEN=-1 THEN GOSUB 9
20 :: GOTO 720
800 CALL POSITION(#28,ZEILE,
SPALTE)
810 IF ZEILE>168 THEN CALL L
OCATE(#28,161,SPALTE):: GOTO
 730
820 CALL COINC(ALL,TREFFEN):
: IF TREFFEN=-1 THEN GOSUB 9
20 :: GOTO 720
830 IF ZEILE<20 THEN GOSUB 9
80 ELSE 860
840 IF P/3=INT(P/3)THEN CALL
 DELSPRITE(#27,#26,#25):: ZA
HL=28 :: TEMPO=TEMPO+3 :: DI
SPLAY AT(1,16):STR$(P/3+1);"
. DURCHGANG" ELSE 720
850 ZEIT=0 :: CALL HCHAR(24,
8,120,10):: GOTO 620
860 IF SPALTE<16 OR SPALTE>2
32 THEN GOSUB 920 :: GOTO 72
0
870 CALL COINC(ALL,TREFFEN):
: IF TREFFEN=-1 THEN GOSUB 9
20 :: GOTO 720
880 GOTO 730
890 !
900 ! *** FEHLER ***
910 !
920 CALL SOUND(500,110,1)::
F=F+1 :: IF F>=10 THEN 1200
930 DISPLAY AT(24,27):STR$(F
)
940 RETURN
950 !
960 ! *** PUNKTE ***
970 !
980 CALL POSITION(#28,ZEILE,
SPALTE)
990 ZAHL=ZAHL-1 :: CALL SPRI
TE(#ZAHL,112,6,10,SPALTE,0,0
):: CALL DELSPRITE(#28)
1000 CALL COINC(ALL,TREFFEN)
:: IF TREFFEN=-1 THEN CALL D
ELSPRITE(#ZAHL):: ZAHL=ZAHL+
1 :: GOSUB 920 :: GOTO 1020
1010 GOSUB 1060 :: P=P+1
1020 RETURN
1030 !
1040 ! *** MUSIK ***
1050 !
1060 FOR J=1 TO ANZAHL :: RE
AD DAUER,TON :: CALL SOUND(2
50*DAUER,TON,2):: NEXT J
1070 CALL SOUND(750,523,2,39
1,7,329,7)
1080 RESTORE 1140
1090 RETURN
1100 DATA 1,329,1,391,1,391,
1,391
1110 DATA 1,349,1,440,2,440
1120 DATA 1,493,1,493,1,440,
1,493
1130 DATA 1,523,1,587,2,659
1140 DATA 1,329,1,391,1,391,
1,391
1150 DATA 1,349,1,440,2,440
1160 DATA 1,493,1,493,1,440,
1,493
1170 !
1180 ! *** SPIELENDE ***
1190 !
1200 CALL DELSPRITE(ALL):: D
ISPLAY AT(8,9)ERASE ALL:"SPI
ELENDE"
1210 IF TASTE=5 THEN 32767
1220 DISPLAY AT(11,1):"PUNKT
E:";P: :"FEHLER:";F :: DISPL
AY AT(18,1):"FUER NEUES SPIE
L": :"ROTE TASTE DRUECKEN":
:"SONST E FUER ENDE DRUECKEN
"
1230 CALL KEY(1,TASTE,ST)::
IF ST=0 THEN 1230
1240 IF TASTE=18 THEN CALL C
LEAR :: RUN 410
1250 IF TASTE=5 THEN 1260 EL
SE 1230
1260 DISPLAY AT(8,9)ERASE AL
L:"VIELEN DANK" :: DISPLAY A
T(13,7):"AUF WIEDERSEHEN": :
"  BIS ZUM NAECHSTEN SPIEL"
:: DISPLAY AT(21,3):"§ 1984
CHRISTIAN DISCH"
1270 RESTORE :: ANZAHL=25 ::
 GOSUB 1060 :: CALL CLEAR ::
 END
```

```
100 REM**********************
110 REM*                    *
120 REM* SCHIFFVERSENKEN    *
130 REM*  Copyright  by     *
140 REM* Thomas  Zschake    *
150 REM*                    *
160 REM* Bearbeitet  vom    *
170 REM*     Team  des      *
180 REM* Aktuell Verlages   *
190 REM*      Muenchen      *
200 REM*                    *
210 REM*Benoetigte Geraete* *
220 REM* TI99/4A  Konsole   *
230 REM*                    *
270 REM* Speicherbelegung   *
280 REM*    6040 Bytes      *
290 REM**********************
295 REM
300 CALL CLEAR
310 PRINT "...***************
*******"
320 PRINT "...*.............
......*"
330 PRINT "...*.SCHIFFE-VERS
ENKEN.*"
340 PRINT "...*.............
......*"
350 PRINT "...*.....GEGEN.DE
N.....*"
360 PRINT "...*.............
......*"
370 PRINT "...*..C.O.M.P.U.T
.E.R..*"
380 PRINT "...*.............
......*"
390 PRINT "...***************
*******": : : : : : : :
400 FOR DELAY=1 TO 1000
410 NEXT DELAY
420 CALL CLEAR
430 PRINT "BEI DIESER VERSIO
N VON ": :
440 PRINT "SCHIFFE-VERSENKEN
 SPIELST DU": :
450 PRINT "GEGEN DEN COMPUTE
R.": :
460 PRINT "NACH DEM DU DEINE
 SCHIFFE": :
470 PRINT "EINGEGEBEN HAST,E
RSCHEINEN": :
480 PRINT "ZWEI FELDER VOR D
IR.": :
490 CALL KEY(0,K,S)
500 IF S=0 THEN 490
510 CALL CLEAR
520 PRINT "DAS LINKE IST DEI
N FELD.": :
530 PRINT "AUF IHM KANNST DU
 DEINE": :
540 PRINT "SCHUESSE VERFOLGE
N.DABEI": :
550 PRINT "WIRD NATUERLICH Z
WISCHEN": :
560 PRINT "TREFFER UND FEHLS
CHUSS": :
570 PRINT "UNTERSCHIEDEN.DIE
S WIRD AUCH": :
580 PRINT "DURCH UNTERSCHIED
LICHE TOENE": :
590 PRINT "ANGEZEIGT.": :
600 CALL KEY(0,K,S)
610 IF S=0 THEN 600
620 CALL CLEAR
630 PRINT "DAS RECHTE ERFUEL
LT DIE": :
640 PRINT "GLEICHE FUNKTION
FUER DEN": :
650 PRINT "COMPUTER.": : : :
 :
660 CALL KEY(0,K,S)
670 IF S=0 THEN 660
680 DIM A(9,9)
690 DIM B(9,9)
700 DIM C(9,9)
710 DIM R(9,9)
720 O2=0
730 O1=0
740 Z5=0
750 Z7=0
760 C6=0
770 Z4=0
780 Y1=0
790 CALL CHAR(120,"000010107
C101")
800 CALL CHAR(121,"AA55AA55A
A55AA55")
810 CALL CLEAR
820 PRINT "WAEHLE NUN BITTE:
": : : :
830 PRINT "E- SCHIFFE EINGEB
EN": :
840 PRINT "V- SCHIFFE VERSEN
KEN": : : : : :
850 CALL KEY(0,K,S)
860 IF S=0 THEN 850
870 IF (K=69)+(K=101)THEN 91
0
880 IF (K=86)+(K=118)THEN 89
0 ELSE 850
890 GOSUB 1200
```

```
900 GOTO 2080
910 CALL CLEAR
920 PRINT "DU KANNST NUN BEL
IEBIG VIELE": :
930 PRINT "SCHIFFE POSITIONI
EREN.": :
940 PRINT "GIB  ZUERST DIE S
ENKRECHTE,": :
950 PRINT "DANN DIE WAAGEREC
HTE": :
960 PRINT "KOORDINATE EIN.":
 :
970 PRINT "WIEVIEL SCHIFFE M
OECHTEST DU": :
980 PRINT "POSITIONIEREN ?":
 :
990 INPUT "SCHIFFE=":W
1000 PRINT : : : : :
1010 IF (W>80)+(W<1)THEN 990
1020 FOR O=1 TO W
1030 INPUT S,T
1040 IF (LEN(STR$(S))>1)+(LE
N(STR$(T))>1)<>0 THEN 1030
1050 IF (S=0)+(T=0)<>0 THEN
1030
1060 A(S,T)=1
1070 NEXT O
1080 REM COMPUTER SETZT SCHI
FFE
1090 RANDOMIZE
1100 FOR L=1 TO W
1110 M=INT(1+9*RND)
1120 N=INT(1+9*RND)
1130 Y1=Y1+1
1140 IF Y1=1 THEN 1160
1150 IF R(M,N)=4 THEN 1110
1160 R(M,N)=4
1170 B(M,N)=1
1180 NEXT L
1190 GOTO 810
1200 REM SPIELFELD
1210 CALL CLEAR
1220 PRINT "GIB NUN BITTE DE
INEN..NAMEN EIN:": : : :
1230 INPUT "NAME:":A$
1240 CALL CLEAR
1250 D$="EDU MUSST "
1260 E$=STR$(W)
1270 F$="ESCHIFFE TREFFEN"
1280 G$=D$&E$&F$
1290 PRINT G$: : : : : : : :
 : : : : : : : : : : : : : :
 :
1300 S=3
1310 S9=4
1320 Z=6
1330 Z7=6
1340 S1=4
1350 Z1=6
1360 S2=5
1370 Z2=6
1380 GOSUB 1500
1390 S=18
1400 S9=19
1410 Z=6
1420 Z7=6
1430 S1=19
1440 Z1=6
1450 S2=20
1460 Z2=6
1470 GOSUB 1500
1480 CALL VCHAR(5,16,88,15)
1490 GOTO 2090
1500 FOR X=0 TO 10
1510 S=S+1
1520 CALL HCHAR(Z,S,88)
1530 NEXT X
1540 FOR X=0 TO 10
1550 S=S+1
1560 CALL HCHAR(Z+10,S-11,88
)
1570 NEXT X
1580 FOR X=0 TO 9
1590 Z1=Z1+1
1600 CALL HCHAR(Z1,S1,88)
1610 NEXT X
1620 FOR X=0 TO 10
1630 Z1=Z1+1
1640 CALL HCHAR(Z1-11,S1+10,
88)
1650 NEXT X
1660 CALL COLOR(12,16,5)
1670 FOR X=1 TO 9
1680 Z2=Z2+1
1690 CALL HCHAR(Z2,S2,126,9)
1700 NEXT X
1710 F=LEN(A$)
1720 FOR K=1 TO F
1730 D$=SEG$(A$,K,1)
1740 J=ASC(D$)
1750 CALL HCHAR(19,5+K,J)
1760 NEXT K
1770 CALL HCHAR(19,21,67)
1780 CALL HCHAR(19,22,79)
1790 CALL HCHAR(19,23,77)
1800 CALL HCHAR(19,24,80)
1810 CALL HCHAR(19,25,85)
1820 CALL HCHAR(19,26,84)
1830 CALL HCHAR(19,27,69)
1840 CALL HCHAR(19,28,82)
```

```
1850 CALL HCHAR(22,7,84)
1860 CALL HCHAR(22,8,82)
1870 CALL HCHAR(22,9,69)
1880 CALL HCHAR(22,10,70)
1890 CALL HCHAR(22,11,70)
1900 CALL HCHAR(22,12,69)
1910 CALL HCHAR(22,13,82)
1920 CALL HCHAR(22,22,84)
1930 CALL HCHAR(22,23,82)
1940 CALL HCHAR(22,24,69)
1950 CALL HCHAR(22,25,70)
1960 CALL HCHAR(22,26,70)
1970 CALL HCHAR(22,27,69)
1980 CALL HCHAR(22,28,82)
1990 FOR X=1 TO 9
2000 S9=S9+1
2010 CALL HCHAR(Z-1,S9,X+48)
2020 NEXT X
2030 FOR X=1 TO 9
2040 Z7=Z7+1
2050 CALL HCHAR(Z7,S1-1,X+48
)
2060 NEXT X
2070 RETURN
2080 REM SPIELER SCHIESST
2090 CALL KEY(0,KEY,STATUS)
2100 IF (KEY<49)+(KEY>57)<>0
 THEN 2090
2110 CALL HCHAR(3,16,60)
2120 IF STATUS=0 THEN 2090
2130 CALL HCHAR(3,6,KEY)
2140 CALL HCHAR(3,8,58)
2150 S=KEY-48
2160 CALL KEY(0,KEY,STATUS)
2170 IF (KEY<49)+(KEY>57)<>0
 THEN 2160
2180 IF STATUS=0 THEN 2160
2190 CALL HCHAR(3,10,KEY)
2200 T=KEY-48
2210 IF B(S,T)=1 THEN 2230
2220 IF B(S,T)<>1 THEN 2250
2230 GOSUB 2270
2240 GOTO 2080
2250 GOSUB 2630
2260 GOTO 2430
2270 REM   TREFFER SPIELER

2280 Q=S+6
2290 Y=T+4
2300 01=01+1
2310 CALL HCHAR(Q,Y,120)
2320 CALL SOUND(+500,131,1)
2330 Z6=Z6+1
2340 IF Z6=10 THEN 2370
2350 CALL HCHAR(22,5,Z6+48)
2360 GOTO 2410
2370 Z4=Z4+1
2380 Z6=0
2390 CALL HCHAR(22,4,Z4+48)
2400 CALL HCHAR(22,5,48)
2410 IF 01=W THEN 2910
2420 RETURN
2430 REM COMPUTER SCHIESST
2440 CALL HCHAR(3,16,62)
2450 RANDOMIZE
2460 S=0
2470 T=0
2480 S=INT(1+9*RND)
2490 T=INT(1+9*RND)
2500 Z5=Z5+1
2510 IF Z5=1 THEN 2530
2520 IF C(S,T)=3 THEN 2450
2530 C(S,T)=3
2540 CALL HCHAR(3,24,S+48)
2550 CALL HCHAR(3,26,58)
2560 CALL HCHAR(3,28,T+48)
2570 IF A(S,T)=1 THEN 2590
2580 IF A(S,T)<>1 THEN 2610
2590 GOSUB 2690
2600 GOTO 2430
2610 GOSUB 2850
2620 GOTO 2080
2630 REM  FEHLSCHUSS SPIELER

2640 E=S+6
2650 D=T+4
2660 CALL HCHAR(E,D,121)
2670 CALL SOUND(+200,523,1)
2680 RETURN
2690 REM  TREFFER COMPUTER

2700 Q=S+6
2710 Y=T+19
2720 02=02+1
2730 CALL HCHAR(Q,Y,120)
2740 CALL SOUND(+500,131,1)
2750 Z8=Z8+1
2760 IF Z8=10 THEN 2790
2770 CALL HCHAR(22,20,Z8+48)
2780 GOTO 2830
2790 Z9=Z9+1
2800 Z8=0
2810 CALL HCHAR(22,19,Z9+48)
2820 CALL HCHAR(22,20,48)
2830 IF 02=W THEN 3030
2840 RETURN
2850 REM  FEHLSCHUSS COMPUTE
R
2860 W2=S+6
2870 P=T+19
```

```
2880 CALL HCHAR(W2,P,121)
2890 CALL SOUND(+200,523,1)
2900 RETURN
2910 FOR DELAY=1 TO 600
2920 NEXT DELAY
2930 CALL CLEAR
2940 F$="BRAVO,"
2950 H$="DU HAST GEWONNENE!"
2960 Z$=F$&A$
2970 PRINT Z$: :
2980 PRINT H$: : : : :
2990 FOR N=1 TO 10
3000 CALL SOUND(-100,110+N^3
,N)
3010 NEXT N
3020 GOTO 3120
3030 FOR DELAY=1 TO 600
3040 NEXT DELAY
3050 CALL CLEAR
3060 J$="TJA,"
3070 I$=J$&A$
3080 PRINT I$: :
3090 PRINT "DU HAST LEIDER V
ERLOREN!": : : :
3100 FOR DELAY=1 TO 600
3110 NEXT DELAY
3120 PRINT "ICH HOFFE ES HAT
 DIR SPASS": :
3130 PRINT "GEMACHT.": : : :
3140 PRINT "......AUF WIEDER
SEHN": : : : : :
```

Fortsetzung von Seite 79

Schließen sich zum Beispiel 3 – 5 Leute zusammen, von denen jeder einen Commodore 64 will, kann man fest damit rechnen, daß ein Händler 5 – 10% Nachlaß auf die Geräte gibt.
In diesem Fall wäre das eine Ersparnis von 30 – 60 DM pro Mitglied!!!
Daß hierbei auch mehr als 5 Leute in einem Ort aufzustöbern möglich ist, läßt sich am Beispiel meiner Klasse belegen, in der sich allein innerhalb der letzten 2 Monate 5 (fünf) Mitschüler einen Commodore 64 gekauft hatten.
Damals hatte jeder für sich allein sein Gerät gekauft, doch beim nächsten Kauf (dem VC 1541-Laufwerk) werden sie und ich = WIR uns zusammenschließen, denn nach einer Anfrage bei einem Commodore-Händler hatte sich nach langem Hin und Her herausgestellt, daß dieser bei einer Abnahme von 5 Laufwerken, einen Preisnachlaß von 8% gewähren würde!
Ebenso konnten wir bei Diskettenabnahmen von 100 Stück auf einmal einen Preis-Nachlaß von 12% erreichen.
Von Vorteil ist hier natürlich, daß wir uns fast jeden Tag in der Schule sehen; man kann sich besser aufeinander abstimmen, wann man was wo kauft.
Es mag zwar sein, daß einige Leser wegen 50 DM keine solchen Anstrengungen unternehmen wollen, aber wenn es sich, wie in unserem Fall, richtiggehend anbietet, ist man selbst schuld, wenn man sein Geld verschenkt.
Also mein Tip:
Bevor man sich entscheidet, eine größere Anschaffung zu machen, sollte man, gerade im Bereich EDV, seine Bekannten fragen, ob diese dasselbe kaufen wollen, oder jemanden aus ihrem Bekanntenkreis kennen, welcher für einen Gemeinschaftskauf in Frage käme.
Seien Sie aber nicht überrascht, wenn Sie auf mehr Mitkäufer treffen, als eigentlich maximal eingeplant.
Es wäre bestimmt auch nicht der erste Computerclub, der aus solch einer Einkaufsgemeinschaft entstehen könnte.

Uwe Fütterer

BREAK-OUT

Dieses Programm ist die TI-Version des gleichnamigen Spielhallenoldies.
Mit Hilfe eines Schlägers und eines Balles müssen aus einer Mauer Steine herausgebrochen werden. Je höher die Steine in der Mauer angeordnet sind, desto mehr Punkte gibt es für einen Treffer. Unterhalb der Mauer befindet sich der Schläger, mit dessen Hilfe der Ball in Richtung Mauer geschlagen wird. Treffen Sie den Ball nicht, so bekommen Sie einen von 6 Bällen abgezogen. Sind alle Bälle verbraucht, ist das Spiel beendet. Gesteuert wird der Schläger mit den Tasten 1,=,Q und P. Die beiden oberen Tasten sind für schnelle und die beiden unteren Tasten für langsame Rechts-Linksbewegungen programmiert. Durch ein Menü kann man die Ein- und Ausfallswinkel des Balles bestimmen.

Reinhold Frieling

PUSH-IT

Dies ist ein Würfelspiel für einen Mann und einen Computer. Jeder der beiden Spieler hat 4 Spielsteine und muß diese, möglichst als Erster, über eine Strecke von 30 Feldern bewegen. Doch so einfach ist es nun wieder nicht. Würfelt man eine ungerade Zahl, so muß man diese Augenzahl rückwärts gehen. Nur mit einer geraden Augenzahl kann man sich dem Ziel nähern. Zieht man auf ein Feld, das schon besetzt ist, so wird dieser Stein nicht geschlagen, sondern um die gleiche Würfelzahl nach vorne oder hinten gesetzt, je nachdem ob die Augenzahl gerade oder ungerade war. Bevor man sich eine Figur zum Ziehen aussuchen kann, müssen zuerst alle Spielsteine ins Spiel gebracht werden.

W. Heimann

```
100 ! ************************
110 ! *                      *
120 ! *      Break Out       *
130 ! *    Copyright by      *
140 ! *      R.Frieling      *
150 ! *                      *
160 ! *   Bearbeitet vom     *
170 ! *       Team des       *
180 ! *  Aktuell Verlages    *
190 ! *       Muenchen       *
200 ! *                      *
210 ! * Benoetigte Geraete   *
220 ! *  TI99/4A  Konsole    *
230 ! *      Ext. Basic      *
240 ! *                      *
270 ! *  Speicherbelegung    *
280 ! *      4057 Bytes      *
290 ! ************************

295 !
300 GOTO 310 :: K :: S :: SC
 :: HI :: RO :: RE :: A :: B
 :: C :: W :: SW :: E$ :: AN
 :: CALL SOUND :: CALL CLEAR
 :: CALL HCHAR :: CALL VCHAR
 :: CALL SPRITE :: CALL DELS
PRITE :: CALL POSITION :: CA
LL COLOR :: CALL SCREEN :: C
ALL MOTION :: CALL KEY !$P-
310 ON BREAK NEXT
320 !
330 !SPIELFELDAUFBAU +FARBE
     +ZEICHENDEFINITION
340 !
350 CALL CLEAR :: CALL SCREE
N(5):: FOR A=1 TO 8 :: CALL
COLOR(A,16,1):: NEXT A :: CA
LL COLOR(9,9,9,10,14,14,11,1
1,11,12,15,15,13,3,3,14,8,8)
360 FOR A=3 TO 8 :: CALL COL
OR(A,16,1):: NEXT A :: CALL
COLOR(9,9,9,10,14,14,11,11,1
1,12,15,15,13,3,3,14,8,8)
370 A$="60F0F06000000000"&RP
T$("0",48):: CALL CHAR(128,A
$):: A$=RPT$("0",28)&"FFFF"&
RPT$("0",28)&"FFFF" :: CALL
CHAR(132,A$):: CALL CHAR(64,
"3C4299A1A199423C")
380 DISPLAY AT(1,7):">>>BREA
KOUT<<<": : : : : :TAB(11)
;"§ 1984": :TAB(4);"BY REINH
OLD FRIELING": :"...4408 DUE
LMEN/RORUP": :TAB(7);"KIRCHS
PIEL 17"
390 DISPLAY AT(18,6):"TEL.:
02548/243": : : :"VORSPANN?
J" :: ACCEPT AT(22,11)SIZE(-
1)VALIDATE("JNjn")BEEP:E$ ::
 IF E$="J" OR E$="j" THEN GO
SUB 860
400 ON WARNING NEXT :: DISPL
AY AT(6,1)ERASE ALL:"SPIELWE
ISE? (1-4) 1" :: ACCEPT AT(6
,19)SIZE(-1)VALIDATE("1234")
BEEP:SW :: RE=5 :: RO=1 :: C
ALL CLEAR
410 CALL HCHAR(1,3,30,28)::
CALL HCHAR(22,3,30,28):: CAL
L VCHAR(2,3,30,20):: CALL VC
HAR(2,30,30,20)
420 DISPLAY AT(23,1):"RESERV
E:";RE;TAB(16);"SCORE:";SC:"
HISCORE:";HI;TAB(16);"ROUND:
";RO
430 FOR A=1 TO 6 :: CALL HCH
AR(A+4,7,88+8*A,20):: NEXT A
440 RANDOMIZE :: A,C=14 :: B
=C*(INT(RND*3)-1):: CALL MAG
NIFY(3):: CALL SPRITE(#1,128
,16,88,128,#2,132,7,145,121)
450 DISPLAY AT(14,3)SIZE(24)
:"PRESS >SPACE< TO BEGIN !"
:: FOR W=1 TO 15 :: NEXT W :
: DISPLAY AT(14,3)SIZE(24):"
" :: FOR W=1 TO 10 :: CALL K
EY(0,K,S):: IF K=32 THEN 470
460 NEXT W :: GOTO 450
470 CALL SOUND(100,262,0,294
,5,330,5):: CALL MOTION(#1,A
,B)
480 !
490 !BALLLENKUNG
500 !
510 CALL POSITION(#1,X,Y)::
IF X>=150 AND A=C THEN 600 E
LSE IF X<=88 AND X>32 THEN I
F Y>28 AND Y<216 THEN 750
520 A=(X<16)*-C+(X>=150)*C+(
(X>15)AND(X<150))*-A :: B=(Y
<32)*-C+C*(Y>224)+((Y>31)AND
(Y<225))*-B :: CALL MOTION(#
1,A,B)
530 !
540 !SCHLAEGERLENKUNG
550 !
560 CALL KEY(0,K,S):: CALL M
OTION(#2,0,(K=49)*40+(K=61)*
-40+(K=81)*10+(K=80)*-10)::
GOTO 510
```

```
570 !
580 !BALL GETROFFEN?
590 !
600 CALL MOTION(#1,0,0,#2,0,
0):: CALL POSITION(#2,X1,Y1)
:: D=Y-Y1 :: IF D<=3 AND D>=
-2 THEN B=-C ELSE IF D>3 AND
 D<=8 THEN B=0 ELSE IF D>8 A
ND D<=14 THEN B=C ELSE 680
610 !
620 !JA
630 !
640 A=-C :: CALL MOTION(#1,A
,B):: CALL SOUND(100,-1,0)::
 CALL SOUND(1,30000,30):: GO
TO 510
650 !
660 !NEIN
670 !
680 CALL SOUND(100,-5,0):: R
E=RE-1 :: IF RE>=0 THEN DISP
LAY AT(23,9)SIZE(7):RE :: GO
TO 440 ELSE CALL DELSPRITE(A
LL):: HI=MAX(HI,SC):: DISPLA
Y AT(24,9)SIZE(7):HI :: SC,A
N=0
690 DISPLAY AT(12,9)SIZE(10)
:"GAME OVER!" :: DISPLAY AT(
14,5)SIZE(18):"PRESS REDO OR
 BACK" :: DISPLAY AT(12,9)SI
ZE(10):"" :: DISPLAY AT(14,5
)SIZE(18):""
700 FOR A=1 TO 10 :: CALL KE
Y(0,K,S):: IF K=6 THEN 400 E
LSE IF K=15 THEN 350
710 NEXT A :: GOTO 690
720 !
730 !STEIN GETROFFEN?
740 !
750 CALL MOTION(#1,0,0,#2,0,
0):: X=INT((X-1)/8)+1 :: Y=I
NT((Y-1)/8)+1 :: CALL GCHAR(
X,Y,T):: IF T=32 THEN CALL M
OTION(#1,A,B):: GOTO 560
760 !
770 !JA
780 !
790 CALL SOUND(100,(11-X)*30
+230,0):: CALL HCHAR(X,Y,32)
:: AN=AN+1 :: SC=SC+(11-X)*1
5*RO :: DISPLAY AT(23,22):SC
800 IF AN<120 THEN 820 ELSE
RO=RO+1 :: AN=0 :: CALL CLEA
R :: CALL DELSPRITE(ALL):: C
ALL SOUND(400,262,0):: CALL
SOUND(400,294,0)
810 CALL SOUND(400,330,0)::
CALL SOUND(400,349,0):: GOTO
 410
820 A=(SW=1)*-C-(SW=2)*C+(SW
=4)*A-(SW=3)*A :: B=(SW=1)*0
-(SW=2)*B-(SW=3)*B-(SW=4)*B
:: CALL MOTION(#1,A,B):: GOT
O 560
830 !
840 !VORSPANN
850 !
860 DISPLAY AT(1,10)ERASE AL
L:"VORSPANN": : :"DER SINN D
ES SPIELES BESTEHT": :"DARIN
,MOEGLICHST VIELE STEI-": :"
NE ZU BEKOMMEN,WOBEI SIE DA-
"
870 DISPLAY AT(10,1):"FUER S
ORGEN MUESSEN,DASS DER": :"B
ALL NICHT AUF DEN BODEN": :"
FAELLT.": : :"JE HOEHER SICH
 DIE STEINE"
880 DISPLAY AT(19,1):"BEFIND
EN,DESTO MEHR PUNKTE": :"BEK
OMMEN SIE DAFUER.": : :TAB(2
7);">>"
890 CALL KEY(0,K,S):: IF S=0
 THEN 890 ELSE DISPLAY AT(1,
1)ERASE ALL:"IHNEN STEHEN 6
BAELLE ZUR": :"VERFUEGUNG.":
 :"NACHDEM SIE DIE SPIELWEIS
E"
900 DISPLAY AT(7,1):"FESTGEL
EGT HABEN,BEGINNT DAS": :"SP
IEL!": : :"SCHLAEGERBEWEGUNG
 MIT": :">1< : SCHNELL NACH
LINKS": :">=< : SCHNELL NACH
 RECHTS"
910 DISPLAY AT(18,1):">Q< :
LANGSAM NACH LINKS": :">P< :
 LANGSAM NACH RECHTS": : : :
"ALLES VERSTANDEN? J" :: ACC
EPT AT(24,19)SIZE(-1)VALIDAT
E("JNjn")BEEP:E$ :: IF E$="J
" OR E$="j" THEN RETURN
920 DISPLAY AT(12,8)ERASE AL
L:"ALSO NOCHMAL!" :: FOR W=1
 TO 200 :: NEXT W :: GOTO 86
0
```

```
100 REM********************
110 REM*                  *
120 REM*     Push-it      *
130 REM*  Copyright  by   *
140 REM* Wolfgang Heimann *
150 REM*                  *
160 REM* Bearbeitet  vom  *
170 REM*    Team  des     *
180 REM* Aktuell Verlages *
190 REM*     Muenchen     *
200 REM*                  *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole *
230 REM*                  *
240 REM* Speicherbelegung *
250 REM*   10134 Bytes    *
260 REM********************
270 REM
280 REM SPIELFELDAUFBAU,VARI
ABLENBELEGUNG,ZEICHENDEFINIT
ION,FARBEN
290 RANDOMIZE
300 DATA 15,3,13,3,11,3,9,3,
7,3,5,3,5,5,5,7,5,9,5,11
310 DATA 5,13,3,13,3,15,3,17
,3,19,3,21,3,23,3,25,3,27,5,
27
320 DATA 7,27,9,27,11,27,13,
27,15,27,15,25,15,23,15,21,1
5,19,15,17
330 DIM SF(31,3),M(30)
340 DEF SZ=SZIEL*2-1
350 DEF CZ=CZIEL*2-1
360 CALL CLEAR
370 B=1
380 CHAUS=4
390 SHAUS=4
400 PRINT "SPIELANLEITUNG.?.
.<J/N>"
410 CALL KEY(0,K,S)
420 IF (S<>0)*((K=74)+(K=78)
+(K=106)+(K=110))THEN 430 EL
SE 410
430 IF (K=78)+(K=110)THEN 46
0
440 CALL CLEAR
450 GOSUB 2930
460 CALL CLEAR
470 PRINT TAB(10);"12.13..15
..17..19": :
480 PRINT "6.7.8.9"
490 PRINT TAB(15);"14..16..1
8"
500 PRINT TAB(13);"11.......
.20"
510 PRINT TAB(9);"10"
520 PRINT "..5.............
.....21": :
530 PRINT "..4.............
.....22": :
540 PRINT "..3...ZIEL......
.....23": :
550 PRINT "..2.............2
9..27....24": :
560 PRINT "..1.............
.........25"
570 PRINT TAB(15);"30..28..2
6": : : : : : :
580 RESTORE 300
590 FOR I=1 TO 30
600 READ KH,KV
610 SF(I,0)=KH
620 SF(I,1)=KV
630 NEXT I
640 CALL VCHAR(15,2,65,2)
650 CALL VCHAR(15,16,90,2)
660 CALL CHAR(96,"0103070F1F
3F7EFC",97,"80C0E0F0F8FC7E3F
",98,"FC7E3F1F0F070301",99,"
3F7EFCF8F0E0C080",104,"00000
00000000303",106,"0303")
670 CALL CHAR(105,"000000000
000C0C",107,"C0C0",108,"0000
3C3C3C3C",109,"",110,"3C3C3C
3C0000C0C0",111,"030300003C3
C3C3C")
680 CALL CHAR(112,"3C3C3C3C0
0000303",113,"C0C000003C3C3C
3C",114,"3C3C3C3C00003C3C",1
15,"3C3C00003C3C3C3C",116,"1
03070FFFF703010")
690 CALL CHAR(117,"000000FFF
F",118,"080C0EFFFF0E0C08",12
0,"",121,"FFFFFFFFFFFFFFFF")
700 CALL SCREEN(8)
710 CALL COLOR(9,12,5,10,2,1
6,11,2,16,12,4,7)
720 CALL HCHAR(1,7,117)
730 CALL HCHAR(23,2,96)
740 CALL HCHAR(23,3,97)
750 CALL HCHAR(24,2,98)
760 CALL HCHAR(24,3,99)
770 CALL HCHAR(23,30,121)
780 CALL HCHAR(23,31,109)
790 CALL HCHAR(24,30,109)
800 CALL HCHAR(24,31,121)
810 P$="DEINE STEINE..MEINE
STEINE"
820 FOR I=1 TO 26
830 CALL HCHAR(24,3+I,ASC(SE
G$(P$,I,1)))
840 NEXT I
850 P$="DEINE EINGABE:....:
MEIN ZUG"
860 FOR I=1 TO 4
870 CALL HCHAR(12+I,7,32,8)
880 NEXT I
890 FOR I=1 TO 28
900 CALL HCHAR(20,2+I,ASC(SE
G$(P$,I,1)))
910 NEXT I
920 CALL VCHAR(5,3,120,12)
930 CALL VCHAR(5,4,120,12)
940 CALL HCHAR(5,5,120,10)
```

```
950 CALL HCHAR(6,5,120,10)
960 CALL HCHAR(3,13,120,16)
970 CALL HCHAR(4,13,120,16)
980 CALL VCHAR(5,27,120,12)
990 CALL VCHAR(5,28,120,12)
1000 CALL HCHAR(15,17,120,10
)
1010 CALL HCHAR(16,17,120,10
)
1020 REM SPIELER ZIEHT
1030 IF B=-1 THEN 1500
1040 IF (SZIEL=4)+(CZIEL=4)T
HEN 3310
1050 CALL SOUND(100,1046,3)
1060 SF(0,2)=1
1070 CALL HCHAR(1,3,32,2)
1080 CALL HCHAR(2,3,32,2)
1090 CALL HCHAR(2,7,83)
1100 WW=INT(RND*4)+3
1110 FOR I=1 TO WW
1120 W=INT(RND*6)+1
1130 GOSUB 2380
1140 NEXT I
1150 IF W/2<>INT(W/2)THEN 11
80
1160 W=W*-1
1170 GOTO 1230
1180 IF SHAUS+SZIEL<>4 THEN
1230
1190 CALL SOUND(100,2000,0,1
000,0)
1200 FOR I=1 TO 500
1210 NEXT I
1220 GOTO 1500
1230 CALL KEY(0,K,S)
1240 IF (S=0)+(K>51)+(K<48)T
HEN 1230
1250 CALL HCHAR(20,18,K)
1260 E1$=CHR$(K)
1270 CALL SOUND(100,1000,0)
1280 CALL KEY(0,K,S)
1290 IF (S=0)+(K>57)+(K<48)T
HEN 1280
1300 CALL HCHAR(20,19,K)
1310 CALL SOUND(100,500,0)
1320 E$=E1$&CHR$(K)
1330 ZUG=VAL(E$)
1340 IF ZUG>30 THEN 1380
1350 IF SF(ZUG,2)<>1 THEN 13
80
1360 IF (ZUG=0)*(W/2<>INT(W/
2))THEN 1380
1370 IF (SHAUS=0)*(ZUG=0)THE
N 1380 ELSE 1410
1380 CALL SOUND(100,250,0)
1390 CALL HCHAR(20,18,32,2)
1400 GOTO 1230
1410 CALL KEY(0,K,S)
1420 IF S=0 THEN 1410
1430 IF K<>13 THEN 1380
1440 IF (SHAUS<>0)*(W/2=INT(
W/2))THEN 1450 ELSE 1480
1450 ZUG=0
1460 SHAUS=SHAUS-1
1470 CALL HCHAR(20,18,48,2)
1480 GOSUB 2050
1490 REM COMPUTER ZIEHT
1500 IF (SZIEL=4)+(CZIEL=4)T
HEN 3310
1510 CALL SOUND(100,523,3)
1520 CALL HCHAR(2,7,67)
1530 CALL HCHAR(1,3,32,2)
1540 CALL HCHAR(2,3,32,2)
1550 WW=INT(RND*4)+3
1560 FOR I=1 TO WW
1570 W=INT(RND*6)+1
1580 GOSUB 2380
1590 NEXT I
1600 SF(0,2)=-1
1610 IF W/2<>INT(W/2)THEN 16
70
1620 W=W*-1
1630 IF CHAUS=0 THEN 1680
1640 ZUG=0
1650 CHAUS=CHAUS-1
1660 GOTO 2020
1670 IF CHAUS+CZIEL=4 THEN 2
040
1680 FOR I=1 TO 30
1690 IF SF(I,2)<>-1 THEN 181
0
1700 FELD=I
1710 BEW=1
1720 IF (FELD-W>30)+(FELD-W<
1)THEN 1780
1730 IF SF(FELD-W,2)=0 THEN
1780
1740 B1=SF(FELD-W,2)
1750 BEW=BEW+(B1*-1)
1760 FELD=FELD-W
1770 GOTO 1720
1780 IF W/2=INT(W/2)THEN 180
0
1790 BEW=BEW*-1
1800 M(I)=BEW
1810 NEXT I
1820 X=1
1830 FOR I=1 TO 30
1840 IF SF(I,2)<>-1 THEN 193
0
1850 ON X GOTO 1860,1900
1860 ZUG=I
1870 MAXI=M(I)
1880 X=2
1890 GOTO 1930
1900 IF M(I)<=MAXI THEN 1930
1910 MAXI=M(I)
1920 ZUG=I
1930 NEXT I
1940 A$=STR$(ZUG)
1950 IF LEN(A$)=2 THEN 1970
1960 A$=CHR$(48)&A$
1970 CALL HCHAR(20,18,ASC(SE
G$(A$,1,1)))
```

```
1980 CALL HCHAR(20,19,ASC(SE
G$(A$,2,1)))
1990 FOR I=1 TO 30
2000 M(I)=0
2010 NEXT I
2020 GOSUB 2050
2030 REM SUB:STEINE SCHIEBEN
2040 GOTO 1040
2050 SF(ZUG,3)=SF(ZUG,2)
2060 SF(ZUG,2)=0
2070 IF (ZUG-W<1)+(ZUG-W>30)
THEN 2110
2080 SF(ZUG-W,3)=SF(ZUG-W,2)
2090 SF(ZUG-W,2)=SF(ZUG,3)
2100 GOTO 2120
2110 GOSUB 2720
2120 IF ZUG=0 THEN 2290
2130 KH=SF(ZUG,0)
2140 KV=SF(ZUG,1)
2150 IF SF(ZUG,2)=0 THEN 222
0
2160 IF SF(ZUG,2)=1 THEN 225
0
2170 CALL HCHAR(KH,KV,121)
2180 CALL HCHAR(KH+1,KV,109)
2190 CALL HCHAR(KH,KV+1,109)
2200 CALL HCHAR(KH+1,KV+1,12
1)
2210 GOTO 2290
2220 CALL HCHAR(KH,KV,120,2)
2230 CALL HCHAR(KH+1,KV,120,
2)
2240 GOTO 2290
2250 CALL HCHAR(KH,KV,96)
2260 CALL HCHAR(KH,KV+1,97)
2270 CALL HCHAR(KH+1,KV,98)
2280 CALL HCHAR(KH+1,KV+1,99
)
2290 ZUG=ZUG-W
2300 IF (ZUG<1)+(ZUG>30)THEN
 2330
2310 IF SF(ZUG+W,3)<>0 THEN
2070
2320 SF(ZUG,2)=SF(ZUG,3)
2330 FOR I=1 TO 30
2340 SF(I,3)=0
2350 NEXT I
2360 RETURN
2370 REM SUB:WUERFEL+PFEILE
MALEN
2380 ON W GOTO 2390,2440,249
0,2540,2570,2620
2390 CALL HCHAR(1,3,104)
2400 CALL HCHAR(2,3,106)
2410 CALL HCHAR(1,4,105)
2420 CALL HCHAR(2,4,107)
2430 GOTO 2650
2440 CALL HCHAR(1,3,108)
2450 CALL HCHAR(1,4,109)
2460 CALL HCHAR(2,3,109)
2470 CALL HCHAR(2,4,108)
2480 GOTO 2680
2490 CALL HCHAR(1,3,104)
2500 CALL HCHAR(1,4,110)
2510 CALL HCHAR(2,3,111)
2520 CALL HCHAR(2,4,107)
2530 GOTO 2650
2540 CALL HCHAR(1,3,108,2)
2550 CALL HCHAR(2,3,108,2)
2560 GOTO 2680
2570 CALL HCHAR(1,3,112)
2580 CALL HCHAR(1,4,110)
2590 CALL HCHAR(2,3,111)
2600 CALL HCHAR(2,4,113)
2610 GOTO 2650
2620 CALL HCHAR(1,3,114,2)
2630 CALL HCHAR(2,3,115,2)
2640 GOTO 2680
2650 CALL HCHAR(1,6,116)
2660 CALL HCHAR(1,8,32)
2670 GOTO 2700
2680 CALL HCHAR(1,6,32)
2690 CALL HCHAR(1,8,118)
2700 RETURN
2710 REM SUB:ZIEL- UND HAUSK
ONTROLLE
2720 IF SF(ZUG,3)=0 THEN 291
0
2730 IF ZUG-W>30 THEN 2790
2740 IF SF(ZUG,3)=1 THEN 277
0
2750 CHAUS=CHAUS+1
2760 GOTO 2910
2770 SHAUS=SHAUS+1
2780 GOTO 2910
2790 IF SF(ZUG,3)=1 THEN 286
0
2800 CZIEL=CZIEL+1
2810 CALL HCHAR(13,14-CZ,121
)
2820 CALL HCHAR(13,15-CZ,109
)
2830 CALL HCHAR(14,14-CZ,109
)
2840 CALL HCHAR(14,15-CZ,121
)
2850 GOTO 2910
2860 SZIEL=SZIEL+1
2870 CALL HCHAR(15,14-SZ,96)
2880 CALL HCHAR(15,15-SZ,97)
2890 CALL HCHAR(16,14-SZ,98)
2900 CALL HCHAR(16,15-SZ,99)
2910 RETURN
2920 REM SUB:SPIELANLEITUNG
2930 PRINT "..PUSH - IT": :
:
2940 PRINT "jeder hat 4 stei
ne, die von A nach Z gezogen
 werden"
2950 PRINT "muessen. mit ein
er geraden..zahl wird vorwae
rts, mit.."
```

```
2960 PRINT "einer ungeraden
rueckwaerts gezogen."
2970 PRINT "im bildschirm wi
rd oben.....links die gewuer
felte zahl,"
2980 PRINT "die zugrichtung
und der am..zug befindliche
spieler"
2990 PRINT "(c=comp.,s=spiel
er) gezeigt.jeder zug muss 2
-stellig"
3000 PRINT "eingegeben (z.b.
:06) und mit<ENTER> bestaeti
gt werden."
3010 PRINT "es muessen nach
moeglichkeitalle steine hera
usgezogen"
3020 PRINT "werden, dies ges
chieht durchdie eingabe 00 .
"
3030 PRINT :"DRUECKE EINE TA
STE !"
3040 CALL KEY(0,K,S)
3050 IF S=0 THEN 3040
3060 CALL VCHAR(1,1,32,768)
3070 PRINT "der computer zie
ht spieler- steine, wenn moe
glich, auch"
3080 PRINT "dann 'raus, wenn
 ein andererzug eingegeben w
urde."
3090 PRINT "eine falsche ein
gabe wird...nicht akzeptiert
."
3100 PRINT "die eingabe kann
 vor <ENTER>durch irgendeine
 taste"
3110 PRINT "geloescht werden
.............ueber die felder
 1 und 30"
3120 PRINT "darf beliebig we
it hinausge-zogen werden."
3130 PRINT "kommt ein stein
auf ein be- setztes feld, wi
rd der dor-"
3140 PRINT "tige stein nicht
 geworfen,..sondern um den g
leichen weg"
3150 PRINT "in die gleiche r
ichtung.....geschubst."
3160 PRINT :"DRUECKE EINE TA
STE !"
3170 CALL KEY(0,K,S)
3180 IF S=0 THEN 3170
3190 CALL VCHAR(1,1,32,768)
3200 PRINT "am ende des spie
ls wird die punktzahl gezeig
t, die sich"
3210 PRINT "aus der zahl der
 felder er- gibt, die der ve
rlierer noch"
3220 PRINT "bis ins ziel bra
uchen wuer- de. die punkte w
erden von "
3230 PRINT "spiel zu spiel a
ddiert."
3240 PRINT :"VIEL SPASS !!"
3250 PRINT :"DRUECKE EINE TA
STE !"
3260 CALL KEY(0,K,S)
3270 IF S=0 THEN 3260
3280 CALL VCHAR(1,1,32,768)
3290 RETURN
3300 REM SIEGER + PUNKTE
3310 B=B*-1
3320 IF CZIEL=4 THEN 3450
3330 P$="DU HAST GEWONNEN !"
3340 GOSUB 3840
3350 REM L.V.BEETHOVEN, SINF
. NR.5, OPUS 67, 2. SATZ
3360 DATA 2,156,1,196,4,208,
4,208,4,233,8,262
3370 DATA 2,208,1,233,4,262,
4,262,4,277,8,311
3380 DATA 2,262,1,277,8,311,
2,262,1,277,8,311
3390 RESTORE 3360
3400 FOR I=1 TO 18
3410 READ TD,TF
3420 CALL SOUND(TD*200,TF,5,
TF*2,0,TF*4,10)
3430 NEXT I
3440 GOTO 3550
3450 P$="DU HAST VERLOREN !"
3460 GOSUB 3840
3470 REM L.V.BEETHOVEN, SINF
. NR.3,OPUS 55 'EROICA', 2.S
ATZ
3480 DATA 1,156,1,156,1,139,
1,156,1,123,1,165,2,139
3490 DATA 1,185,1,185,1,208,
1,233,1,247,1,156,2,139
3500 RESTORE 3480
3510 FOR I=1 TO 14
3520 READ TD,TF
3530 CALL SOUND(TD*1000,TF,0
,TF*2,5,TF*4,10)
3540 NEXT I
3550 FOR I=1 TO 30
3560 IF SF(I,2)=0 THEN 3580
3570 P=P+31-I
3580 NEXT I
3590 P=P+(CHAUS*31)+(SHAUS*3
1)
3600 IF SZIEL=4 THEN 3620
3610 P=P*-1
3620 SCORE=SCORE+P
3630 S$=STR$(SCORE)
3640 P$="DEINE PUNKTE : "&S$
3650 GOSUB 3840
3660 P=0
3670 CZIEL=0
```

```
3680 SZIEL=0
3690 CHAUS=4
3700 SHAUS=4
3710 FOR I=1 TO 30
3720 SF(I,2)=0
3730 SF(I,3)=0
3740 NEXT I
3750 CALL SOUND(3000,300,30)
3760 CALL SOUND(333,333,30)
3770 P$="NEUES SPIEL ?..<J/N
>"
3780 GOSUB 3840
3790 CALL KEY(0,K,S)
3800 IF (S<>0)*((K=74)+(K=10
6)+(K=110)+(K=78))THEN 3810
ELSE 3790
3810 IF (K=74)+(K=106)THEN 8
50
3820 END
3830 REM SUB:$-PLAZIERUNG IN
 ZEILE 20
3840 CALL HCHAR(20,1,32,32)
3850 FOR I=1 TO LEN(P$)
3860 CALL HCHAR(20,2+I,ASC(S
EG$(P$,I,1)))
3870 NEXT I
3880 RETURN
```

„MONOPOLY“

Das Programm Monopoly ist ein Versuch, das bekannte Brettspiel auf den Ti-99/4A umzusetzen. Es gelten soweit als möglich die Spielregeln des Originals.

Nach dem Start mit RUN zeichnet der Computer zunächst das Spielbrett auf den Bildschirm.

In der Mitte zwischen den Straßen wird nach der Anzahl der Spieler und deren Namen gefragt.

Danach beginnt das eigentliche Spiel: es wird reihum mit der Taste „w“ gewürfelt, wobei die jeweilige Figur um die gewürfelte Zahl vorgesetzt wird. Ist das betreffende Feld eine Straße, so wird zunächst geprüft, ob sie schon verkauft ist. Wenn ja, dann schreibt der Computer, wem wieviel Miete gezahlt werden muß.

Ist die Straße noch nicht verkauft, so kann man wählen zwischen: kaufen, nicht kaufen und einer Übersicht, die durch Drücken der Taste „u“ aufgerufen wird. Diese Übersicht gibt die Miete der Straße in allen Bebauungszuständen an.

Drückt man die Taste „u“, wenn der Computer zum Würfeln auffordert, erhält man die Nummern der Straßen, die man besitzt. Die Badstr. trägt beispielsweise die Nummer 1, die Parkstr. die Nummer 25.

Desweiteren wird bei jedem Würfeln der momentane Kontostand angezeigt.

Wenn alle Straßen verkauft sind, kann gebaut werden. Ausnahmen bilden dabei das E-werk, das Wasserwerk und der Bahnhof, auf denen nicht gebaut werden kann.

Hat man sich entschieden, wo man bauen möchte, gibt man die Straßennummer ein, die vom Computer auf Gültigkeit geprüft wird. Will man nicht bauen, gibt man eine „0“ ein. Ist die Straßennummer eingegeben, wird gefragt, wieviele Häuser gebaut werden sollen. Möglich sind 4 Häuser, eine 5 bedeutet, daß man ein Hotel wünscht. Die Bebauung wird jeweils neben der Straße angezeigt.

Die Kaufpreise für ein Haus betragen:

- in der unteren Reihe 1000,– DM
- in der linken Reihe 2000,– DM
- in der oberen Reihe 3000,– DM
- in der rechten Reihe 4000,– DM

Nach Eingabe der Hauszahl erwartet der Computer ein „j“ oder ein „n“, je nachdem, ob Sie mit Bauen fertig sind. Geben Sie also ein „j“ ein, ist der nächste Spieler an der Reihe.

Wollen Sie später noch einmal bauen, geben sie ein „b“ ein, wenn Sie zum Würfeln aufgefordert werden.

Außer den Straßen gibt es noch die Ereignis- und Gemeinschaftsfelder, bei denen irgendwelche Ereignisse geschildert werden, die Geld bringen, aber auch kosten können.

Kommt man auf das Feld „frei parken“, so erhält man alles Geld, das bisher durch Steuern oder Strafen bezahlt werden mußte. Wenn man ins Gefängnis muß, kann man sich mit einer Kaution von 1000,– DM freikaufen. Ansonsten kommt man nach der 3. Runde frei. Als Gehalt erhält man jedesmal, wenn man über Los kommt, 4000,– DM.

Das Spiel ist beendet, wenn einer der Spieler kein Geld mehr hat.

T. Lukassen

```
100 ! **********************
110 ! *                    *
120 ! *      MONOPOLY      *
130 ! *                    *
140 ! *    Copyright by    *
150 ! *                    *
160 ! *  Torsten Lukassen  *
170 ! *                    *
180 ! *                    *
190 ! * Benoetigte Geraete *
200 ! *  TI99/4A  Konsole  *
210 ! *     Ext. Basic     *
220 ! *                    *
260 ! *  Speicherbelegung  *
270 ! *    10470 Bytes     *
280 ! *                    *
290 ! **********************
300 DIM NAME$(4),PO(4),M(26)
,ZEI$(8),G(4),BE(26),KA(26),
KN(4)
310 BA=0 :: FOR I=1 TO SP ::
 G(I)=30000 :: KN(I)=0 :: NE
XT I :: DR=1 :: GOSUB 1040 :
: GOSUB 1050
320 CALL CLEAR :: CALL SCREE
N(16):: FOR I=1 TO 14 :: CAL
L COLOR(I,7,1):: NEXT I
330 DATA FF809E909C909E80,FF
809EA0AEA29C80,FF809C8888889
C80,FF80BEA2A2A2BE80,FF80A2A
2AAAA9480,FF80808080808080
340 DATA 809E909C909080FF,80
A2B2AAA6A280FF,80848C84848E8
0FF,808C8284888E80FF,80868A9
29F8280FF,8080808080808080FF
350 DATA 80F8A4B8A4F8808,809
E909C909E8080,809EA0AEA29C80
80,007C4444447C0000,00103010
10380000,00180408101C0000
360 DATA 0038041C0438000,000
40C143E0400,003C203C043C00,0
03C203C243C00,003808081C0800
,003C243C243C00
370 DATA 003C243C043C00,8090
9090909E8080,80A2B2AAA6A2808
0,809EA09C82BC8080,809C828E8
29C8080,80868A929F828080
380 DATA 0149516151490101,01
89C9A999890101,0078242424780
0FF,003C2038203C00FF,003C405
C443800FF,00080808080800FF
390 DATA 00404040407800FF,00
4464544C4400FF,003C403804780
0FF,007C4444447C00FF,0028081
0203800FF,001828487C0800FF
400 DATA 00000000000000FF,FF
0038447C444400,FF00782438247
800,FF00784070407800,FF00784
070404000,FF003C405C443800
410 DATA FF0044447C444400,FF
0038101010380O,FF00446C54444
400,FF004464544C4400,FF007C4
444447C00,FF00784478404000
420 DATA FF00784478484400,FF
003C4038047800,FF007C1010101
000,FF00444444443800,FF,0101
0101010101FF,FF0139457D45450
1
430 DATA FF01391111113901,80
9E909E829E8080,0101010101010
101
440 DATA 0038447C4444,007824
382478,003840404038,00782424
2478,007840704078,0078407040
40
450 DATA 003C405C4438,004444
7C4444,003810101038,00080808
4830,002428302824,0040404040
78
460 DATA 00446C544444,004464
544C44,007C4444447C,00784478
4040,003844544834,0078447848
44
470 DATA 003C40380478,007C10
101010,004444444438,00444428
2810,004444545428
480 DATA 004428102844,004428
101010,007C0810207C,FF010101
01010101,809E909C90908080,80
BCA2BCA2A18080,FF00242830282
4
490 DATA 8080808080808080
500 RESTORE 330 :: FOR I=33
TO 127 :: READ B$ :: CALL CH
AR(I,B$):: NEXT I
510 CALL VCHAR(1,2,96,24)::
CALL VCHAR(1,31,127,24)
520 ZEI$(1)="OWN^SZXN!WNR$VN
W%LXX"&CHR$(34)&"UNY#TX[par?
umst/nis;pla.rwe.st"&CHR$(32
)&"/efaBFK\44oo!eld>8oo}k _
6oo(CFG"
530 ZEI$(2)="wien&"&RPT$("["
,18)&"{hauperst"&CHR$(127)&R
PT$(" ",18)&"'tst 4ooo"&CHR$
(127)&RPT$(" ",18)&"'6ooo"
540 ZEI$(3)="PNSN"&CHR$(127)
&RPT$(" ",18)&"'PNSNinsc"&CH
R$(127)&RPT$(" ",18)&"'insch
aft"&CHR$(127)&RPT$(" ",18)&
"'haft"
550 ZEI$(4)="MLQT"&CHR$(127)
&RPT$(" ",18)&"'MLQThof "&CH
R$(127)&RPT$(" ",18)&"'hofs4
ooo"&CHR$(127)&RPT$(" ",18)&
"'64oo"
560 ZEI$(5)="QLON"&CHR$(127)
&RPT$(" ",18)&"'XYNZnst "&CH
R$(127)&RPT$(" ",18)&"'er  2
8oo"&CHR$(127)&RPT$(" ",18)&
"'2ooo"
```

```
570 ZEI$(6)="N[[[["&CHR$(127)
&RPT$(" ",18)&"'NWNRwerk"&CH
R$(127)&RPT$(" ",18)&"'gnis3
ooo"&CHR$(127)&RPT$(" ",18)&
"'feld"
580 ZEI$(7)="XNN["&CHR$(127)
&RPT$(" ",18)&"'VLW~st  "&CH
R$(127)&RPT$(" ",18)&"'st  2
8oo,"&RPT$("K",18)&"\7ooo"
590 ZEI$(8)="PNO]post.rei.li
s<teu-ad &[[[eng@st  /nis.ns
t.r  <t  :os DGK\IJHH'BEA*HH
H+HHH)IHH,KKK"
600 CALL SPRITE(#1,109,5,32,
64,#2,111,5,32,90,#3,110,5,3
2,120,#4,111,5,32,152)
610 CALL SPRITE(#5,112,5,56,
88,#6,111,5,56,120,#7,108,5,
56,152,#8,121,5,56,184)
620 FOR I=1 TO 8 :: DISPLAY
AT(3*I-2,1):ZEI$(I):: NEXT I
630 DISPLAY AT(20,7)SIZE(14)
:"von t lukassen" :: FOR I=1
 TO 200 :: NEXT I :: DISPLAY
 AT(10,7)SIZE(16):"wieviele
spieler" :: DISPLAY AT(12,7)
SIZE(9):"maximal 4"
640 ACCEPT AT(12,18)SIZE(1)V
ALIDATE("1234"):SP :: GOSUB
1540 :: DISPLAY AT(10,7)SIZE
(13):"gib bitte die" :: DISP
LAY AT(12,7)SIZE(9):"namen e
in"
650 FOR I=1 TO SP :: G(I)=30
000 :: KN(I)=0 :: ACCEPT AT(
12+2*I,7)VALIDATE("abcdefghi
jklmnopqrstuvwxyz ")BEEP SIZ
E(5):NAME$(I):: NEXT I
660 CALL CHAR(142,"3C7E3C183
C7EFF")
670 DATA 2,5,11,13
680 RESTORE 670 :: I=0
690 I=I+1 :: READ FA :: IF I
>2 THEN XI=228 ELSE XI=212
700 IF I=2 OR I=4 THEN YI=18
6 ELSE YI=170
710 CALL SPRITE(#10+I,142,FA
,YI,XI):: IF I<SP THEN 690
720 IF DR=2 OR DR=4 THEN YI=
186 ELSE YI=170
730 IF DR>2 THEN XI=228 ELSE
 XI=212
740 IF KN(DR)THEN 1180 ELSE
GOSUB 1540 :: DISPLAY AT(10,
7)SIZE(15):NAME$(DR):: DISPL
AY AT(12,7)SIZE(15):"muss wu
erfeln" :: GOSUB 1390
750 CALL PATTERN(#10+DR,32):
: CALL KEY(0,K,S):: CALL PAT
TERN(#10+DR,142):: IF K<>119
 AND K<>117 AND K<>98 THEN 7
50
760 SPR=2 :: IF K=117 THEN 1
310 ELSE IF K=98 THEN BA=SP-
1
770 RANDOMIZE :: W1=INT(RND*
6)+1 :: ON W1 GOSUB 970,980,
990,1000,1010,1020
780 CALL CHAR(141,A$):: CALL
 SPRITE(#9,141,2,120,160)
790 W2=INT(RND*6)+1 :: ON W2
 GOSUB 970,980,990,1000,1010
,1020
800 CALL CHAR(143,A$):: CALL
 SPRITE(#10,143,2,120,176)::
 PO(DR)=PO(DR)+W1+W2
810 IF PO(DR)<7 THEN CALL LO
CATE(#10+DR,YI,XI-PO(DR)*32)
:: GOTO 840 ELSE IF PO(DR)<1
4 THEN CALL LOCATE(#10+DR,YI
-(PO(DR)-6)*24,XI-192):: GOT
O 840
820 IF PO(DR)<19 THEN CALL L
OCATE(#10+DR,YI-168,XI-192+(
PO(DR)-13)*32):: GOTO 840 EL
SE IF PO(DR)<26 THEN CALL LO
CATE(#10+DR,YI-144+(PO(DR)-2
0)*24,XI):: GOTO 840
830 PO(DR)=PO(DR)-26 :: G(DR
)=G(DR)+4000 :: GOTO 810
840 IF BE(PO(DR))=0 AND PO(D
R)<>2 AND PO(DR)<>4 AND PO(D
R)<>11 AND PO(DR)<>13 AND PO
(DR)<>15 AND PO(DR)<>6 AND P
O(DR)<>19 AND PO(DR)<>21 THE
N 850 ELSE 860
850 IF BE(PO(DR))=0 AND PO(D
R)<>23 AND PO(DR)<>24 AND PO
(DR)<>0 THEN 1440
860 IF PO(DR)=2 OR PO(DR)=23
 THEN 1210
870 IF PO(DR)=4 OR PO(DR)=15
 OR PO(DR)=24 THEN 1240
880 IF PO(DR)=11 OR PO(DR)=2
1 THEN GOTO 1280
890 IF PO(DR)=8 OR PO(DR)=17
 THEN 1070 ELSE IF PO(DR)=10
 THEN 1120 ELSE IF PO(DR)=13
 THEN 1150 ELSE IF PO(DR)=6
OR PO(DR)=0 THEN 920
900 IF PO(DR)=19 THEN 1160
910 IF BE(PO(DR))THEN 1420 E
LSE 840
920 FOR I=1 TO 26 :: IF BE(I
)THEN 930 ELSE GOSUB 1550
930 NEXT I
940 BA=BA+1 :: IF BA>SP THEN
 950 ELSE GOSUB 1580
950 IF G(DR)<0 THEN 1350 ELS
E DR=DR+1 :: IF DR>SP THEN D
R=1
960 GOTO 720
970 A$="0000001818" :: RETUR
N
```

```
980 A$="COC0000000000303" ::
 RETURN
990 A$="COC0001818000303" ::
 RETURN
1000 A$="C3C300000000C3C3" :
: RETURN
1010 A$="C3C300181800C3C3" :
: RETURN
1020 A$="DBDB00000000DBDB" :
: RETURN
1030 DATA 1200,4000,2000,0,2
400,0,2800,3000,2800,4000,0,
4000,0,4400,0,4800,3000,5600
,0,6000,0,6400,0,0,7000,0
1040 RESTORE 1030 :: FOR I=1
 TO 26 :: READ KA(I):: NEXT
I :: RETURN
1050 DATA 80,0,120,0,160,0,2
00,0,200,0,0,280,0,360,0,400
,0,440,0,520,0,580,0,0,700,0
1060 RESTORE 1050 :: FOR I=1
 TO 26 :: READ M(I):: NEXT I
 :: RETURN
1070 IF BE(PO(DR))=0 THEN 14
40 ELSE IF BE(PO(DR))=DR THE
N 920
1080 IF BE(8)=BE(17)THEN FAK
=200 ELSE FAK=80
1090 M(8),M(17)=FAK*(W1+W2)
1100 GOSUB 1540 :: DISPLAY A
T(12,7)SIZE(15):"zahle ";NAM
E$(BE(PO(DR))):: DISPLAY AT(
14,6)SIZE(15):M(PO(DR));"dm"
1110 G(DR)=G(DR)-M(PO(DR)) ::
 G(BE(PO(DR)))=G(BE(PO(DR)))
+M(PO(DR)):: GOTO 1270
1120 IF BE(PO(DR))=0 THEN 14
40 ELSE IF BE(PO(DR))=DR THE
N 920
1130 GOSUB 1540 :: DISPLAY A
T(12,7)SIZE(12):"zahle 500 d
m" :: DISPLAY AT(14,7)SIZE(1
0):"an ";NAME$(BE(PO(DR)))
1140 G(DR)=G(DR)-500 :: G(BE
(10))=G(BE(10))+500 :: GOTO
1270
1150 GOSUB 1540 :: DISPLAY A
T(10,7)SIZE(14):"du erhaelst
 dm" :: DISPLAY AT(12,6)SIZE
(6):PR :: G(DR)=G(DR)+PR ::
PR=0 :: GOTO 1270
1160 GOSUB 1540 :: DISPLAY A
T(10,7)SIZE(8):"gehe ins" ::
 DISPLAY AT(12,7)SIZE(10):"g
efaengnis" :: CALL LOCATE(#1
0+DR,YI,XI-192):: PO(DR)=6 :
: KN(DR)=1
1170 CALL KEY(0,K,S):: IF S=
0 THEN 1170
1180 IF KN(DR)=4 THEN 1200 E
LSE GOSUB 1540 :: DISPLAY AT
(10,7)SIZE(12):"willst du dm
" :: DISPLAY AT(12,7)SIZE(12
):"1000 kaution" :: DISPLAY
AT(14,7)SIZE(6):"zahlen" ::
DISPLAY AT(16,7)SIZE(3):"j n
"
1190 CALL KEY(0,K,S):: IF S=
0 THEN 1190 ELSE IF K=106 TH
EN G(DR)=G(DR)-1000 :: PR=PR
+1000 :: KN(DR)=0 :: GOTO 74
0
1200 KN(DR)=KN(DR)+1 :: IF K
N(DR)=4 THEN KN(DR)=0 :: GOS
UB 1540 :: DISPLAY AT(10,7)S
IZE(12):"du bist frei" :: GO
TO 920 ELSE 920
1210 GOSUB 1540 :: IF PO(DR)
=2 THEN FP=4000 ELSE FP=2000
1220 DISPLAY AT(10,7)SIZE(13
):"du musst";FP :: DISPLAY A
T(12,7)SIZE(10):"dm steuern"
 :: DISPLAY AT(14,7)SIZE(8):
"bezahlen"
1230 G(DR)=G(DR)-FP :: PR=PR
+FP :: GOTO 1270
1240 GOSUB 1540 :: ON INT(RN
D*2)+1 GOTO 1250,1260
1250 DISPLAY AT(10,7)SIZE(14
):"ruecke bis auf" :: DISPLA
Y AT(12,7)SIZE(7):"los vor"
:: PO(DR)=26 :: CALL LOCATE(
#10+DR,YI,XI):: GOTO 1270
1260 DISPLAY AT(10,7)SIZE(12
):"betrunken im" :: DISPLAY
AT(12,7)SIZE(6):"dienst" ::
DISPLAY AT(14,7)SIZE(13):"st
rafe dm 400" :: G(DR)=G(DR)-
400 :: PR=PR+400 :: GOTO 127
0
1270 CALL KEY(0,K,S):: IF S=
0 THEN 1270 ELSE 920
1280 GOSUB 1540 :: ON INT(RN
D*2)+1 GOTO 1290,1300
1290 DISPLAY AT(10,7)SIZE(11
):"bank irrtum" :: DISPLAY A
T(12,7)SIZE(11):"du erhaelst
" :: DISPLAY AT(14,7)SIZE(7)
:"dm 4000" :: G(DR)=G(DR)+40
00 :: GOTO 1270
1300 DISPLAY AT(10,7)SIZE(16
):"du erbst dm 2000" :: G(DR
)=G(DR)+2000 :: GOTO 1270
1310 GOSUB 1540 :: DISPLAY A
T(10,7)SIZE(5):NAME$(DR):: D
ISPLAY AT(12,7)SIZE(16):"dir
 gehoeren die" :: DISPLAY AT
(14,7)SIZE(11):"strassen nr"
```

```
1320 TR=0 :: TI=0 :: FOR I=1
 TO 25 :: IF BE(I)=DR THEN D
ISPLAY AT(16+TI*2,7+TR*3)SIZ
E(3):I :: TR=TR+1 :: IF TR>4
 THEN TR=0 :: TI=TI+2
1330 NEXT I
1340 CALL KEY(0,K,S):: IF S
THEN IF SPR=1 THEN 1580 ELSE
 740 ELSE 1340
1350 CALL CLEAR :: CALL DELS
PRITE(ALL):: CALL CHARSET ::
 DISPLAY AT(2,1):NAME$(DR);"
hat kein geld":"mehr. damit
ist das spiel...beendet."
1360 IF SP>2 THEN DISPLAY AT
(6,1):"der sieger ergibt sic
h nun  aus eurem kapital"
1370 FOR I=1 TO SP :: DISPLA
Y AT(10+I*2,1):NAME$(I);" ha
t";G(I);"dm" :: NEXT I
1380 FOR I=1 TO 10000 :: NEX
T I
1390 DISPLAY AT(19,7)SIZE(11
):"du hast nun" :: DISPLAY A
T(20,7)SIZE(10):"dm";G(DR)::
 RETURN
1400 IF I=0 THEN FT=1 ELSE I
F I=1 THEN FT=5 ELSE IF I=2
THEN FT=15 ELSE IF I=3 THEN
FT=30 ELSE IF I=4 THEN FT=40
 ELSE FT=52
1410 RETURN
1420 GOSUB 1540 :: IF BE(PO(
DR))=DR THEN 920 ELSE DISPLA
Y AT(10,7)SIZE(16):NAME$(BE(
PO(DR)));" bekommt" :: DISPL
AY AT(12,6)SIZE(16):M(PO(DR)
);"dm miete"
1430 G(BE(PO(DR)))=G(BE(PO(D
R)))+M(PO(DR)):: G(DR)=G(DR)
-M(PO(DR)):: GOTO 1270
1440 GOSUB 1540
1450 IF PO(DR)=8 OR PO(DR)=1
0 OR PO(DR)=17 THEN DISPLAY
AT(10,7)SIZE(16):"willst du
kaufen" :: DISPLAY AT(12,7)S
IZE(3):"j n" ELSE 1480
1460 CALL KEY(0,K,S):: IF S=
0 THEN 1460 :: IF K=106 THEN
 1470 ELSE IF K=110 THEN 920
 ELSE 1460
1470 BE(PO(DR))=DR :: G(DR)=
G(DR)-KA(PO(DR)):: GOSUB 139
0 :: GOTO 1270
1480 GOSUB 1540 :: DISPLAY A
T(12,7)SIZE(13):"willst du d
ie" :: DISPLAY AT(14,7)SIZE(
14):"strasse kaufen" :: DISP
LAY AT(16,7)SIZE(5):"j n u"
1490 CALL KEY(0,K,S):: IF K=
106 THEN 1500 ELSE IF K=117
THEN 1510 ELSE IF K=110 THEN
 920 ELSE 1490
1500 BE(PO(DR))=DR :: G(DR)=
G(DR)-KA(PO(DR)):: GOSUB 139
0 :: GOTO 1270
1510 GOSUB 1540 :: DISPLAY A
T(10,6)SIZE(18):"die miete b
etraegt" :: DISPLAY AT(12,6)
SIZE(11):"allein" :: DISPLAY
 AT(12,16)SIZE(8):M(PO(DR));
"dm" :: FOR I=1 TO 5 :: DISP
LAY AT(I+12,6)SIZE(9):"mit";
I;"hau"
1520 GOSUB 1400 :: D=INT(FT*
M(PO(DR))):: DISPLAY AT(I+12
,16)SIZE(8):D;"dm" :: NEXT I
 :: DISPLAY AT(17,6)SIZE(9):
"mit 1 hot"
1530 CALL KEY(0,K,S):: IF S=
0 THEN 1530 ELSE 1440
1540 FOR I=10 TO 20 :: DISPL
AY AT(I,6)SIZE(18):"........
.................." :: NE
XT I :: CALL DELSPRITE(#9,#1
0):: RETURN
1550 IF I<>2 AND I<>4 AND I<
>6 AND I<>10 AND I<>11 AND I
<>13 AND I<>15 AND I<>19 AND
 I<>21 AND I<>23 AND I<>24 A
ND I<>26 THEN 950 ELSE RETUR
N
1560 J=127 :: FOR I=49 TO 52
 :: CALL CHARPAT(I,BI$):: J=
J+1 :: CALL CHAR(J,BI$):: NE
XT I :: CALL CHARPAT(53,BI$)
:: CALL CHAR(136,BI$):: CALL
 COLOR(13,15,13,14,15,7)
1570 RETURN
1580 GOSUB 1540 :: DISPLAY A
T(10,7)SIZE(15):"du kannst"
:: DISPLAY AT(12,7)SIZE(15):
"jetzt bauen" :: DISPLAY AT(
14,7)SIZE(6):NAME$(DR)
1590 FOR I=1 TO 25 :: IF KA(
I)>5 THEN KA(I)=0
1600 NEXT I :: GOSUB 1390
1610 DISPLAY AT(16,7)SIZE(2)
:"wo" :: ACCEPT AT(16,12)SIZ
E(2)VALIDATE(DIGIT,"u"):WO$
:: IF WO$="u" THEN SPR=1 ::
GOTO 1310 ELSE WO=VAL(WO$)::
 IF WO>25 THEN 1610
1620 IF WO=0 THEN 950 ELSE I
F BE(WO)<>DR OR WO=8 OR WO=1
0 OR WO=17 THEN 1610
1630 IF WO<6 THEN KP=1000 EL
SE IF WO<13 THEN KP=2000 ELS
```

```
E IF WO<19 THEN KP=3000 ELSE
 KP=4000
1640 DISPLAY AT(18,7)SIZE(8)
:"wieviel" :: ACCEPT AT(18,1
5)SIZE(1)VALIDATE("012345"):
NR
1650 IF KA(WO)+NR<=5 THEN I=
KA(WO):: GOSUB 1400 :: M(WO)
=M(WO)/FT :: KA(WO)=KA(WO)+N
R ELSE GOTO 1580
1660 GOSUB 1560 :: IF KA(WO)
<=4 THEN KR=127+KA(WO)ELSE K
R=136
1670 IF WO<6 THEN CALL HCHAR
(21,29-WO*4,KR):: GOTO 1690
ELSE IF WO<13 THEN CALL HCHA
R(23-(WO-6)*3,7,KR):: GOTO 1
690 ELSE IF WO<19 THEN CALL
HCHAR(4,4+(WO-13)*4,KR):: GO
TO 1690
1680 CALL HCHAR(2+(WO-19)*3,
26,KR)

1690 I=KA(WO):: GOSUB 1400 :
: M(WO)=INT(FT*M(WO)):: G(DR
)=G(DR)-KP*NR
1700 CALL KEY(0,K,S):: IF K=
106 THEN RETURN ELSE IF K=11
0 THEN 1580 ELSE 1700
```

WORTSUCHSPIEL

WORTSUCHSPIEL

Haben Sie schon einmal versucht, ein Kreuzworträtsel selbst zu bauen? Dann wissen Sie ja auch, wie kompliziert das ist. Haben Sie waagrecht alle Wörter in die Kästchen geschrieben, passen die Wörter senkrecht nicht richtig oder in Waagrecht sitzt ein Füllfeld, welches nun senkrecht im Wege ist. Lassen Sie sich nicht entmutigen, sondern übertragen Sie diese Aufgabe einfach Ihrem TI.

Geben Sie ihm die Größe des Rätselfeldes an und die Wörter, die Sie für das Rätsel benutzen möchten. Alles andere macht er für Sie. Hat er die Rätselfläche gefüllt, können Sie am Monitor versuchen, Ihre vorgegebenen Wörter wieder zu finden oder Sie können das ganze Rätsel über einen Drucker ausgeben lassen. Bauen Sie sich so Ihre eigene Rätselzeitschrift.

STEUERUNG DES PROGRAMMES

Nach dem Laden des Programmes werden Sie nach der Größe des Rätselfeldes gefragt. Als Vorgabe wurde ein Feld der Größe 20 x 20 gewählt. Wünschen Sie eine andere Größe, so ändern Sie bitte diese Vorlage um. Es sind quadratische oder rechteckige Felder möglich. Nun können Sie das erste Wort eingeben und danach durch –ENTER– bestätigen. Am unteren Bildschirmrand blinken nun die Statements –PROC'D–, –BEGIN– und –REDO– auf.

Durch drücken von FCTN PROC'D übernimmt der Computer dieses Wort für das Rätsel und Sie können das nächste Wort eingeben. Möchten Sie nichts mehr eingeben, so drücken Sie nach dem letzten Wort –FCTN BEGIN–.

Durch –FCTN REDO– können Sie an dem zuletzt eingegebenen Wort Änderungen vornehmen oder es komplett löschen.

Nach Eingabe des letzten Wortes müssen Sie dem Computer im Dialog angeben, auf welches Gerät er das fertige Rätsel ausgeben soll. Hier haben Sie 3 Wahlmöglichkeiten:

1. Pio-Drucker
2. Bildschirm
3. Anderes Gerät

Bei Nr. 3 geben Sie z. Beispiel an: Dsk1. Raetsel oder CS1. Danch beginnt der TI mit dem Einsortieren der Wörter, wobei er Ihnen auf dem Monitor anzeigt, welches Wort er gerade bearbeitet. Je größer Sie das Feld gewählt haben, desto länger wird nun Ihre Geduld in Anspruch genommen. Nachdem das Feld gefüllt ist, bekommen Sie das fertige Rätsel auf das von Ihnen gewählte Geräte ausgegeben.

Wenn Sie die Lösung wünschen, drücken Sie –ENTER– und der Computer zeigt Ihnen das Feld ohne Füllbuchstaben.

Nochmaliges Drücken von –ENTER– beendet das Programm.

P. Ullrich

CODO

CODO läuft auf der Grundkonfiguration des TI 99/4A. Der Planet CODO wird ständig von Zyklopenraumschiffen bedroht. Dies macht es erforderlich, daß jeder Schüler des Planeten CODO auf einem Simulator die Abwehr der Zyklopen übt.

Die zur Abwehr bereitstehenden Laserkanonen werden von einem Roboter bedient, welcher mit dem Joystick gesteuert wird. Um einen Schuß auszulösen, muß die Aktivierungstaste so lange gedrückt werden, bis der Erkennungssound ertönt. Der Abschußwinkel beträgt bei waagrechter Auslenkung 45°, bei senkrechter Auslenkung 90°.

Gelingt es einem Zyklopenraumschiff so nahe zu kommen, daß es den Planeten beschießen kann, so landet kurz darauf ein Zyklop unterhalb den Lasern. Es ist nun höchste Alarmstufe, weil er die Energiereserven vernichtet. Findet er keine Energie, beschießt er die Felder über sich und somit auch den kleinen CODO. Im Falle eines Treffers führt dies zur Selbstvernichtung des Zyklopen und den Verlust von fünf Energieeinheiten. Aber am besten wird der kleine CODO über die schützende Energiereserve dirigiert. Nun heißt es, auf den Zyklopen zu warten und im richtigen Moment die Aktivierungstaste zu drücken. Sollte die Energie, inklusive Reserven unter zehn Einheiten absinken, so unterbricht der Computer das Spiel und es ertönt eine kleine Melodie. Werden mehr als achtzig Punkte erreicht, wird der Schwierigkeitsgrad erhöht.

Durch besonders erfolgreichen Beschuß kann der Energieverlust vermindert werden.

Michael Ehrenschneider

```
100 ! **********************
110 ! *                    *
120 ! *   Wortsuchspiel    *
130 ! *    Copyright by    *
140 ! *   Peter  Ullrich   *
150 ! *                    *
160 ! *   Bearbeitet vom   *
170 ! *      Team des      *
180 ! *  Aktuell Verlages  *
190 ! *      Muenchen      *
200 ! *                    *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole  *
230 ! *     Ext. Basic     *
240 ! *  Drucker optional  *
250 ! *                    *
260 ! *  Speicherbelegung  *
270 ! *     3732 Bytes     *
280 ! **********************

290 !
300 OPTION BASE 1
310 DIM F$(20,30),A$(20),B$(
20),L(20),M$(20,20)
320 ! MIT RAM-ERWEITERUNG DI
E DIMENSIONEN BEI BEDARF ENT
SPRECHEND AENDERN
330 CALL SCREEN(12):: DISPLA
Y AT(1,3)ERASE ALL:"***...WO
RTSUCHSPIEL...***":".."&RPT$
("=",25)
340 DISPLAY AT(7,1):"FELDGRO
ESSE?...20.*.20"
350 ACCEPT AT(7,16)SIZE(-2)V
ALIDATE(DIGIT)BEEP:W$ :: IF
W$="" THEN 350 ELSE IA=VAL(W
$)
360 IF IA<7 OR IA>20 THEN 35
0
370 ACCEPT AT(7,21)SIZE(-2)V
ALIDATE(DIGIT)BEEP:W$ :: IF
W$="" THEN 370 ELSE IE=VAL(W
$)
380 IF IE<7 OR IE>20 THEN 37
0
390 IMAX=MIN(20,MAX(IA,IE))
400 DISPLAY AT(10,1):"GEBEN
SIE BITTE DIE WOERTER EIN."
410 FOR I=1 TO MIN(IA,IE)
420 DISPLAY AT(13,1):"WORT N
R.";I :: DISPLAY AT(24,1):""
430 ACCEPT AT(16,1)SIZE(IMAX
)VALIDATE(UALPHA):A$(I):: IF
 A$(I)="" OR POS(A$(I)," ",1
)>0 THEN 430 ELSE B$(I)=A$(I
):: CALL BACK(B$(I)):: L(I)=
LEN(A$(I))
440 DISPLAY AT(24,1):">> PRO
C'D" :: CALL PROMPT(T):: DIS
PLAY AT(24,1):"" :: IF T=12
THEN 480 ELSE IF T=14 THEN 4
90 ELSE IF T=6 THEN 420
450 DISPLAY AT(24,11):">> BE
GIN" :: CALL PROMPT(T):: DIS
PLAY AT(24,1):"" :: IF T=12
THEN 480 ELSE IF T=14 THEN 4
90 ELSE IF T=6 THEN 420
460 DISPLAY AT(24,21):">> RE
DO" :: CALL PROMPT(T):: DISP
LAY AT(24,1):"" :: IF T=12 T
HEN 480 ELSE IF T=14 THEN 49
0 ELSE IF T=6 THEN 420
470 GOTO 440
480 NEXT I :: I=I-1
490 N=I
500 CALL HCHAR(4,1,32,32*21)
510 DISPLAY AT(7,1):"AUSGABE
EINHEIT:..1":"<1> PIO-DRUCKE
R":"<2> BILDSCHIRM":"<3> AND
ERES GERAET"
520 ACCEPT AT(7,18)SIZE(-1)V
ALIDATE("123"):O
530 IF O=1 THEN DEV$="PIO" :
: GOTO 560
540 IF O=2 THEN O=0 :: GOTO
570
550 DISPLAY AT(14,1):"NAME:"
 :: ACCEPT AT(14,6)BEEP:DEV$
560 OPEN #O:DEV$,OUTPUT
570 CALL HCHAR(4,1,32,32*21)
580 DISPLAY AT(9,1):"IN ARBE
IT ...": : : :"EINGEORDNET S
IND":".O..";"VON";N;"WOERTER
N."
590 DATA 1,0,0,1,-1,0,0,-1,1
,1,-1,1,-1,-1,1,-1
600 DATA 1,-1,-1,-1,-1,1,1,1
,0,-1,-1,0,0,1,1,0
610 CALL SORT(A$(),B$(),L(),
N):: ZF=0
620 FOR ZUG=1 TO N
630 RANDOMIZE
640 FOR J=1 TO IA :: FOR K=1
 TO IE :: M$(J,K)="" :: NEXT
 K :: NEXT J :: ZM=0
650 X=INT(RND*IA+1):: Y=INT(
RND*IE+1)
660 IF M$(X,Y)<>"" OR F$(X,Y
)<>"" THEN 650 ELSE M$(X,Y)=
"@" :: ZM=ZM+1
670 IF ZUG>N/2 THEN RESTORE
600 ELSE RESTORE 590
680 FOR I=1 TO 8
690 READ DX,DY
700 R=INT(RND*2+1):: ON R GO
TO 710,730
710 CALL CHECK(F$(,),A$(ZUG)
,L(ZUG),X,Y,DX,DY,FLAG,IA,IE
,ZF)
720 IF FLAG THEN 800 ELSE IF
 R=2 THEN 760
730 CALL CHECK(F$(,),B$(ZUG)
,L(ZUG),X,Y,DX,DY,FLAG,IA,IE
,ZF)
```

```
740 IF FLAG THEN 800 ELSE IF
 R=2 THEN 710
750 !
760 NEXT I
770 IF ZF+ZM<IA*IE THEN 650
780 DISPLAY AT(22,1)BEEP:"ME
HR KOENNEN NICHT EINGEORD-NE
T WERDEN!"
790 GOTO 840
800 DISPLAY AT(14,1)SIZE(4)B
EEP:ZUG
810 FOR I=1 TO 30 :: CALL KE
Y(0,T,ST):: IF T=15 THEN 840
820 NEXT I
830 NEXT ZUG :: ZUG=ZUG-1
840 PRINT #0: : :
850 IF O THEN BL$=" " :: C$=
"¦" :: PRINT #0:RPT$("-",3+2
*IA)ELSE BL$,C$=""
860 FOR I=1 TO IE :: PRINT #
0:C$;
870 FOR J=1 TO IA
880 IF F$(I,J)="" THEN CH$=C
HR$(INT(RND*26)+65)ELSE CH$=
F$(I,J)
890 PRINT #0:BL$&CH$;
900 NEXT J :: PRINT #0:BL$&C
$
910 NEXT I :: IF O THEN PRIN
T #0:RPT$("-",2*IA+3)
920 IF O=0 THEN 1050
930 CALL ALPHASORT(A$(),N)::
 PRINT #0: : : :"Die zu such
enden Woerter sind:": :
940 FOR I=1 TO ZUG-1 :: PRIN
T #0:A$(I)&", ";
950 NEXT I :: PRINT #0:A$(ZU
G)&"."
960 PRINT #0: : : : : :"Die
Loesung lautet:": :
970 IF O THEN PRINT #0:RPT$(
"-",1+2*IA)
980 FOR I=1 TO IE
990 FOR J=1 TO IA
1000 IF F$(I,J)="" THEN CH$=
" " ELSE CH$=F$(I,J)
1010 PRINT #0:C$&CH$;
1020 NEXT J :: PRINT #0:C$
1030 NEXT I :: IF O THEN PRI
NT #0:RPT$("-",1+2*IA)
1040 CALL KEY(0,K,S):: IF S=
0 THEN 1040 ELSE CALL CLEAR
:: END
1050 CALL KEY(0,K,S):: IF S
THEN 960 ELSE 1050
1060 SUB BACK(STRING$)
1070 C$="" :: DEF BACK$(X)=S
EG$(STRING$,X,1)
1080 FOR J=LEN(STRING$)TO 1
STEP -1
1090 C$=C$&BACK$(J):: NEXT J
 :: STRING$=C$
1100 SUBEND
1110 SUB PROMPT(T)
1120 FOR I=1 TO 8 :: CALL KE
Y(0,T,ST):: IF ST THEN SUBEX
IT
1130 NEXT I
1140 SUBEND
1150 SUB SORT(A$(),B$(),L(),
N)
1160 FOR I=1 TO N :: FOR J=I
+1 TO N
1170 IF L(I)<L(J)THEN CALL S
WAP(A$(I),A$(J)):: CALL SWAP
(B$(I),B$(J)):: CALL SWAP@(L
(I),L(J))
1180 NEXT J :: NEXT I
1190 SUBEND
1200 SUB SWAP(X$,Y$)
1210 Z$=X$ :: X$=Y$ :: Y$=Z$
1220 SUBEND
1230 SUB SWAP@(X,Y)
1240 Z=X :: X=Y :: Y=Z
1250 SUBEND
1260 SUB CHECK(F$(,),C$,L1,X
,Y,DX,DY,FLAG,IA,IE,ZF)
1270 Q,R,FLAG=0
1280 IF R=L1 THEN 1340 ELSE
R=R+1 :: X=X+DX :: Y=Y+DY
1290 IF X<1 OR X>IA OR Y<1 O
R Y>IE THEN SUBEXIT
1300 IF F$(X,Y)<>"" THEN 132
0
1310 GOTO 1280
1320 IF F$(X,Y)=SEG$(C$,R,1)
THEN Q=Q+1 ELSE SUBEXIT
1330 GO TO 1280
1340 FLAG=1
1350 ZF=ZF+L1-Q
1360 F$(X,Y)=SEG$(C$,R,1)
1370 R=R-1 :: X=X-DX :: Y=Y-
DY
1380 IF R THEN 1360
1390 SUBEND
1400 SUB ALPHASORT(A$(),N)
1410 FOR I=1 TO N :: FOR J=I
+1 TO N
1420 IF A$(I)>A$(J)THEN CALL
 SWAP(A$(I),A$(J))
1430 NEXT J :: NEXT I
1440 SUBEND
```

```
100 REM*********************
110 REM*        CODO        *
120 REM*   COPYRIGHT BY     *
130 REM* M.Ehrenschneider    *
160 REM*                    *
180 REM*BENOETIGTE GERAETE*
190 REM* TI 99/4A KONSOLE  *
200 REM*    DSK oder CS     *
210 REM*                    *
220 REM* SPEICHERBELEGUNG  *
230 REM*    10904 BYTES     *
240 REM*                    *
250 REM*********************
255 REM
260 CALL CLEAR
270 RANDOMIZE
280 CALL SCREEN(2)
290 CALL COLOR(1,2,2)
300 FOR Z=2 TO 16
310 CALL COLOR(Z,16,16)
320 NEXT Z
330 C=38.5
340 E=8
350 GOSUB 420
360 GOSUB 420
370 E=9
380 GOSUB 420
390 E=8
400 GOSUB 420
410 GOTO 480
420 FOR Z=1 TO E
430 READ A,B
440 C=INT(C+3.5)
450 CALL HCHAR(A,B,C)
460 NEXT Z
470 RETURN
480 FOR Z=1 TO 16
490 D=INT(RND*14)+3
500 CALL SOUND(100,Z*200,10)
510 CALL COLOR(INT(RND*15)+2
,D,D)
520 FOR Z1=1 TO 30
530 NEXT Z1
540 NEXT Z
550 DATA 16,10,16,9,16,8,17,
8,18,8,19,8,19,9,19,10
560 DATA 17,12,17,13,17,14,1
8,12,18,14,19,12,19,13,19,14
570 DATA 17,16,17,17,17,18,1
6,18,18,18,19,18,19,17,19,16
,18,16
580 DATA 17,20,17,21,17,22,1
8,20,18,22,19,20,19,21,19,22
590 CALL CLEAR
600 REM schrift
610 FOR Z=3 TO 8
620 CALL COLOR(Z,16,2)
630 NEXT Z
640 C$="H0206schwierigkeitsg
rad  TASTE"
650 GOSUB 4330
660 FOR Z=9 TO 12
670 CALL COLOR(Z,4,2)
680 NEXT Z
690 FOR Z=1 TO 2
700 CALL COLOR(Z,2,2)
710 NEXT Z
720 CALL COLOR(13,2,2)
730 CALL VCHAR(3,4,128,13)
740 FOR Z=1 TO 5
750 READ A,B,C
760 CALL HCHAR(A,B,128,C)
770 NEXT Z
780 DATA 1,4,3,2,4,1,3,4,3,1
4,5,5,15,5,5
790 FOR Z=1 TO 9
800 READ A,B,C
810 CALL VCHAR(A,B,43,C)
820 NEXT Z
830 FOR Z=1 TO 4
840 READ A,B,C
850 CALL VCHAR(A,B,43,C)
860 NEXT Z
870 CALL COLOR(2,4,4)
880 DATA 10,10,13,9,11,14,8,
12,15,7,13,18,7,14,18,7,15,1
6,7,16,16,7,17,16,7,18,18
890 DATA 7,19,18,8,20,15,9,2
1,14,10,22,13
900 FOR Z=1 TO 6
910 READ A,B,C
920 CALL HCHAR(A,B,136,C)
930 NEXT Z
940 DATA 10,13,2,10,18,2,11,
13,2,11,18,2,12,16,1,14,14,5
950 CALL COLOR(14,9,9)
960 FOR Z=1 TO 8
970 CALL HCHAR(3+Z*2,28,ASC(
STR$(Z)))
980 FOR Z1=1 TO 15
990 CALL KEY(0,K,S)
1000 IF S<>0 THEN 1050
1010 NEXT Z1
1020 CALL HCHAR(3+Z*2,28,32)
1030 NEXT Z
1040 GOTO 960
1050 CALL HCHAR(3+Z*2,28,32)
1060 CALL COLOR(13,4,4)
1070 KLASSE=Z
1080 PUNKTE=0
1090 RESTORE 1130
1100 FOR Z=1 TO KLASSE
1110 READ L1,L2
1120 NEXT Z
1130 DATA 3,9,4,8,4,6,4,5,5,
5,5,4,6,4,6,3
1140 CALL CHAR(120,"3C7E66DF
DFE7FF66")
1150 CALL COLOR(12,16,2)
1160 FOR Z=4 TO 30
1170 CALL HCHAR(2,Z,120)
1180 CALL HCHAR(2,Z,32)
```

```
1190 NEXT Z
1200 CALL CLEAR
1210 CALL SCREEN(4)
1220 HI=5
1230 E=30
1240 C=18
1250 P=0
1260 FOR Z=3 TO 4
1270 CALL COLOR(Z,12,HI)
1280 NEXT Z
1290 FOR Z=1 TO 6
1300 ZG(Z)=0
1310 NEXT Z
1320 REM  munition
1330 CALL COLOR(16,2,HI)
1340 CALL CHAR(152,"18181818
18181818")
1350 CALL HCHAR(24,2,152,29)
1360 REM  befestigung
1370 FOR Z=3 TO 27 STEP 4
1380 CALL HCHAR(22,Z,144,3)
1390 NEXT Z
1400 CALL CHAR(144,"00000000
000066FF")
1410 CALL COLOR(15,10,HI)
1420 REM  rand
1430 CALL HCHAR(1,1,33,32)
1440 CALL HCHAR(2,1,33,32)
1450 CALL VCHAR(2,1,33,23)
1460 CALL VCHAR(2,31,33,23)
1470 CALL VCHAR(2,32,33,23)
1480 CALL COLOR(1,HI,HI)
1490 REM  raumbasen
1500 FOR Z=2 TO 30 STEP 4
1510 CALL HCHAR(22,Z,136)
1520 NEXT Z
1530 RS$(1)="8060783C3E1F0FF
F"
1540 RS$(2)="18183C3C3C7E7EF
F"
1550 RS$(3)="01061E3C7CF8F0F
F"
1560 CALL CHAR(136,RS$(2))
1570 CALL COLOR(14,4,HI)
1580 REM  laser
1590 CALL CHAR(128,"80402010
08040201")
1600 CALL CHAR(129,"18181818
18181818")
1610 CALL CHAR(130,"01020408
10204080")
1620 CALL COLOR(13,2,HI)
1630 REM  codo
1640 CALL HCHAR(23,18,120)
1650 CALL COLOR(12,16,HI)
1660 REM  zyklobraumschiff
1670 ZZ$="3C7E819999817E3C"
1680 ZN$="182C66C3C3662C18"
1690 ZW$="1268234814286421"
1700 CALL COLOR(10,16,HI)
1710 CALL COLOR(11,16,HI)
1720 CALL CHAR(119,ZW$)
1730 REM  zyklob
1740 CALL CHAR(118,"183C66C3
DBC3FF66")
1750 FOR Z=104 TO 116 STEP 2
1760 CALL CHAR(Z,ZZ$)
1770 NEXT Z
1780 FOR Z=105 TO 117 STEP 2
1790 CALL CHAR(Z,ZZ$)
1800 NEXT Z
1810 FOR Z=5 TO 8
1820 CALL COLOR(Z,16,HI)
1830 NEXT Z
1840 C$="H0104PUNKTE"
1850 GOSUB 4330
1860 C$="H0120ENERGIE"
1870 GOSUB 4330
1880 C$="v0432TEXSAS INSTRUM
ENTS"
1890 GOSUB 4330
1900 C$="H0117"&STR$(KLASSE)
1910 GOSUB 4330
1920 REM  **hauptprogramm**
1930 FOR Z0=1 TO L1
1940 FOR Z1=1 TO L2
1950 CALL JOYST(1,X,Y)
1960 CALL KEY(1,K,S)
1970 IF X=0 THEN 2020
1980 CALL HCHAR(23,C,32)
1990 IF (C+X<>-2)+(C+X<>34)=
-1 THEN 2010
2000 C=C+X
2010 CALL HCHAR(23,C,120)
2020 IF S=0 THEN 2470
2030 CALL SOUND(100,700,11)
2040 E=E-1
2050 GOSUB 3620
2060 CALL GCHAR(22,C,CGV)
2070 IF CGV<>136 THEN 2470
2080 CALL JOYST(1,X,Y)
2090 IF (X<>0)+(Y<>0)=0 THEN
 2080
2100 IF Y<>4 THEN 2130
2110 A=2
2120 GOTO 2210
2130 IF X<>-4 THEN 2160
2140 A=1
2150 GOTO 2210
2160 IF X<>4 THEN 2190
2170 A=3
2180 GOTO 2210
2190 CALL SOUND(-50,-8,9)
2200 GOTO 2080
2210 CALL CHAR(136,RS$(A))
2220 ON A GOTO 2230,2320,239
0
2230 FOR Z=1 TO 19
2240 CALL GCHAR(22-Z,C-Z,CGV
)
```

```
2250 IF CGV=32 THEN 2280
2260 IF (CGV<104)+(CGV>117)=
0 THEN 2550
2270 IF CGV=33 THEN 2470
2280 CALL HCHAR(22-Z,C-Z,128
)
2290 CALL HCHAR(22-Z,C-Z,32)
2300 NEXT Z
2310 GOTO 2470
2320 FOR Z=21 TO 2 STEP -1
2330 CALL GCHAR(Z,C,CGV)
2340 IF (CGV<104)+(CGV>117)=
0 THEN 2550
2350 CALL HCHAR(Z,C,129)
2360 CALL HCHAR(Z,C,32)
2370 NEXT Z
2380 GOTO 2470
2390 FOR Z=1 TO 19
2400 CALL GCHAR(22-Z,C+Z,CGV
)
2410 IF CGV=32 THEN 2440
2420 IF (CGV<104)+(CGV>117)=
0 THEN 2550
2430 IF CGV=33 THEN 2470
2440 CALL HCHAR(22-Z,C+Z,130
)
2450 CALL HCHAR(22-Z,C+Z,32)
2460 NEXT Z
2470 NEXT Z1
2480 GOSUB 2770
2490 NEXT ZO
2500 M=M+1
2510 E=E+INT(P/(M*(7/3)))
2520 GOTO 1930
2530 REM *hauptprogrammend*

2540 REM   treffer ?
2550 CGV=INT(CGV/2)*2
2560 CALL SOUND(-300,880,25,
-3,25)
2570 CALL CHAR(CGV,ZW$)
2580 CALL CHAR(CGV+1,ZW$)
2590 B=CGV-102-((CGV-102)/2)
2600 ZG(B)=0
2610 CALL HCHAR(ZY(B),ZX(B)-
1,32,3)
2620 CALL VCHAR(ZY(B)-1,ZX(B
),32,3)
2630 CALL CHAR(CGV,ZZ$)
2640 CALL CHAR(CGV+1,ZN$)
2650 P=P+6-ZG(ZO)
2660 C$="H0111"&STR$(P)
2670 IF KLASSE=8 THEN 2740
2680 IF P<80 THEN 2740
2690 CALL CLEAR
2700 C$="H1203SIE HABEN ES G
ESCHAFFT"
2710 KLASSE=KLASSE+1
2720 GOSUB 4330
2730 GOTO 1080
2740 GOSUB 4330
2750 GOTO 2470
2760 REM angreifersteuerung
2770 ON ZG(ZO)+1 GOSUB 2790,
2840,2880,2920,3130
2780 RETURN
2790 ZY(ZO)=INT(16*RND)+4
2800 ZX(ZO)=INT(27*RND)+3
2810 CALL HCHAR(ZY(ZO),ZX(ZO
),102+ZO*2)
2820 ZG(ZO)=1
2830 RETURN
2840 CALL HCHAR(ZY(ZO)-1,ZX(
ZO),103+ZO*2)
2850 CALL HCHAR(ZY(ZO)+1,ZX(
ZO),103+ZO*2)
2860 ZG(ZO)=2
2870 RETURN
2880 CALL HCHAR(ZY(ZO),ZX(ZO
)-1,103+ZO*2)
2890 CALL HCHAR(ZY(ZO),ZX(ZO
)+1,103+ZO*2)
2900 ZG(ZO)=3
2910 RETURN
2920 D=ZY(ZO)+2
2930 CALL SOUND(-50,330,4,22
0,4)
2940 CALL SOUND(-50,330,4,44
0,4)
2950 CALL VCHAR(D,ZX(ZO),129
,22-D)
2960 CALL GCHAR(22,ZX(ZO),CG
V)
2970 IF CGV<>144 THEN 3040
2980 CALL HCHAR(22,ZX(ZO),11
9)
2990 E=E-1
3000 GOSUB 3620
3010 CALL HCHAR(22,ZX(ZO),14
4)
3020 CALL VCHAR(D,ZX(ZO),32,
24-D)
3030 GOTO 3050
3040 CALL HCHAR(22,ZX(ZO),11
9)
3050 CALL GCHAR(24,ZX(ZO),CG
V)
3060 CALL VCHAR(D,ZX(ZO),32,
24-D)
3070 IF CGV<>152 THEN 3110
3080 E=E-1
3090 GOSUB 3620
3100 CALL HCHAR(24,ZX(ZO),11
9)
3110 ZG(ZO)=4
3120 RETURN
3130 IF ZX(ZO)>=18 THEN 3160
3140 ZD=1
3150 GOTO 3170
3160 ZD=-1
```

```
3170 CALL HCHAR(22,ZX(ZO),11
9)
3180 E=E-1
3190 GOSUB 3620
3200 CALL HCHAR(24,ZX(ZO),11
8)
3210 CALL HCHAR(22,ZX(ZO),14
4)
3220 ZB=ZX(ZO)
3230 CALL JOYST(1,X,Y)
3240 CALL KEY(1,K,S)
3250 IF X=0 THEN 3290
3260 CALL HCHAR(23,C,32)
3270 C=C+X/2
3280 CALL HCHAR(23,C,120)
3290 IF S=0 THEN 3340
3300 E=E-1
3310 GOSUB 3620
3320 CALL GCHAR(24,C,CGV)
3330 IF CGV=118 THEN 3500
3340 CALL HCHAR(24,ZB,32)
3350 CALL GCHAR(24,ZB+ZD,CGV
)
3360 IF CGV<>33 THEN 3380
3370 ZD=ZD*-1
3380 ZB=ZB+ZD
3390 IF CGV=152 THEN 3470
3400 CALL GCHAR(23,ZB,CGV)
3410 IF CGV<>120 THEN 3450
3420 E=E-5
3430 GOSUB 3620
3440 GOTO 3500
3450 CALL HCHAR(23,ZB,129)
3460 CALL HCHAR(23,ZB,119)
3470 CALL HCHAR(24,ZB,118)
3480 CALL SOUND(-100,-3,7)
3490 GOTO 3230
3500 CALL HCHAR(22,ZX(ZO),14
4)
3510 CALL HCHAR(24,C,119)
3520 CALL VCHAR(ZY(ZO)-1,ZX(
ZO),32,3)
3530 CALL HCHAR(ZY(ZO),ZX(ZO
)-1,32,3)
3540 ZG(ZO)=0
3550 CALL HCHAR(23,C,32)
3560 C=18
3570 CALL HCHAR(18,C,32)
3580 CALL HCHAR(24,Z,32)
3590 CALL HCHAR(23,C,120)
3600 RETURN
3610 REM auswertung
3620 IF E<10 THEN 3670
3630 FOR Z3=1 TO 2
3640 CALL HCHAR(1,28+Z3,ASC(
SEG$(STR$(E),Z3,1)))
3650 NEXT Z3
3660 GOTO 4320
3670 C$="V0901RESERVE !"
3680 GOSUB 4330
3690 FOR Z3=2 TO 30
3700 CALL SOUND(-100,Z3*50+3
50,5)
3710 CALL GCHAR(24,Z3,CGV)
3720 IF CGV<>152 THEN 3750
3730 CALL HCHAR(24,Z3,32)
3740 E=E+1
3750 NEXT Z3
3760 CALL VCHAR(1,1,33,24)
3770 IF E>10 THEN 3620
3780 C$="H0114END"
3790 GOSUB 4330
3800 FOR Z=1 TO 5
3810 CALL COLOR(1,4,4)
3820 CALL COLOR(1,HI,HI)
3830 NEXT Z
3840 RESTORE 3890
3850 FOR Z=1 TO 28
3860 READ D,T
3870 CALL SOUND(D*0.75,T,8)
3880 NEXT Z
3890 DATA 250,440,250,494,25
0,523,250,523,250,523,250,52
3,250,523,125,523,125,523,12
5,587,125,523,250,523
3900 DATA 250,494,250,494,50
0,20000,250,494,250,494,250,
494,250,494,250,440,250,440,
500,20000,250,440,250,392
3910 DATA 250,349,125,392,12
5,349,125,330
3920 CALL CLEAR
3930 CALL COLOR(2,12,HI)
3940 NOTE=INT((6-P/40)*10)/1
0
3950 IF NOTE>=1 THEN 3970
3960 NOTE=1
3970 C$="H0810"&"ZEUGNIS   K
LASSE "&STR$(KLASSE)
3980 GOSUB 4330
3990 C$="H1003LIEBER CODOSCH
UELER IHRE LEIS"
4000 GOSUB 4330
4010 C$="H1203TUNG WIRD MIT
DER NOTE "
4020 GOSUB 4330
4030 C$="H1228"&STR$(NOTE)
4040 GOSUB 4330
4050 C$="H1403HONORIERT"
4060 GOSUB 4330
4070 IF NOTE>4 THEN 4180
4080 IF KLASSE+1<>9 THEN 412
0
4090 C$="H1603SIE HABEN DAS
ABI GESCHAFFT"
4100 GOSUB 4330
4110 STOP
4120 C$="H1603SIE HABEN DAS
ZIEL ERREICHT"
4130 KLASSE=KLASSE+1
4140 GOSUB 4330
```

```
4150 GOTO 4250
4160 C$="H1603SIE NICHT GESC
HAFFT"
4170 GOSUB 4330
4180 IF KLASSE-1>0 THEN 4220
4190 C$="H1603SUCHEN SIE EIN
E ANDERE SCHULE"
4200 GOSUB 4330
4210 STOP
4220 C$="H1603ZIEL NICHT ERR
EICHT"
4230 GOSUB 4330
4240 KLASSE=KLASSE-1
4250 C$="H2003NEUES SCHULJAH
R BEGINNT"
4260 GOSUB 4330
4270 C$="H2203EINE TASTE DRU
ECKEN"
4280 GOSUB 4330
4290 CALL KEY(0,K,S)
4300 IF S=0 THEN 4290
4310 GOTO 1080
4320 RETURN
4330 L=VAL(SEG$(C$,2,2))
4340 S=VAL(SEG$(C$,4,2))
4350 IF SEG$(C$,1,1)="H" THE
N 4410
4360 FOR Z=6 TO LEN(C$)
4370 CALL HCHAR(L,S,ASC(SEG$
(C$,Z,Z)))
4380 L=L+1
4390 NEXT Z
4400 RETURN
4410 FOR Z=6 TO LEN(C$)
4420 CALL HCHAR(L,S,ASC(SEG$
(C$,Z,1)))
4430 S=S+1
4440 NEXT Z
4450 RETURN
```

GOLF

Nach dem Start mit "RUN" erscheint zunächst für einige Sekunden ein Einleitungsbild. Danach fragt der Computer nach der Anzahl der Spieler und deren Namen. Anschließend kann man sich aussuchen, ob man die ganzen 18 Bahnen eines Golfplatzes durchspielen will, oder es bei weniger bewenden läßt. Bei der nächsten Frage wird entschieden, ob die Ziffern auf dem Bildschirm dargestellt werden sollen oder nicht. Die Geheimeingabe hat den Vorteil, daß man seine eigenen Eingaben vor den Mitspielern geheimhalten kann, wodurch eine größere Chancengleichheit gewährt ist. Ein weiterer Beitrag zur Chancengleichheit ist die Eingabe des Handicaps S, das für jeden Spieler einzeln eingegeben wird. Je höher das eingegebene Handicap ist, desto unberechenbarer wird die Bahn des Balles.
Nachdem all diese Eingaben gemacht worden sind, beginnt das eigentliche Spiel. Der Computer fragt zunächst nach dem Schläger, von dem die mögliche Reichweite des Schlages abhängt. Man hat dabei die Wahl zwischen einem "2W", bei dem es sich um einen Holzschläger mit großer Reichweite handelt und zwischen den Eisenschlägern "3I", "5I", "7I", "9I", wobei der letztere die geringste Reichweite hat.
Ist diese Eingabe vorgenommen, muß die Schlagstärke eingegeben werden, die sich ebenfalls auf die Reichweite auswirkt. Zum Schluß wird die Richtung abgefragt. Dieses geschieht, indem man einen Winkel zwischen 0 und 360 Grad eintippt. Um das ganze zu erschweren, wird in dem Kästchen am linken oberen Bildschirmrand ein Strich gezeichnet, an dem der gewünschte Winkel im Uhrzeigersinn angelegt werden muß.
Hat man einmal eine falsche Eingabe gemacht, so kann man diese mit Hilfe der Taste "=" zurücknehmen. Der Computer springt dann zur vorhergegangenen Abfrage zurück.
Als weitere Informationen erhält der Spieler neben dem Kästchen Angaben über die Bahnlänge, die sich natürlich in jeder Runde ändert, und über die durchschnittlich zu brauchende Schlagzahl für jede der 18 Bahnen, genannt Par. Daneben erscheinen die Angaben für jeden der maximal 3 Spieler. (Von oben nach unten): Gesamtschlagzahl, Ort, an dem sich der Ball gerade befindet und die Punktzahl, die man auf der aktuellen Bahn erreicht hat.
Als kleines Extraproblem sind die Bäume, Seen und Bunker für die Spieler gedacht, die sich darin verirren. In diesem Fall verkürzt sich nämlich die Flugbahn erheblich, so daß unter Umständen mehrere Schläge notwendig sind, um das Hindernis zu verlassen.
Sollte es einmal passieren, daß der Ball das Spielfeld verläßt, wird der Ball an den Spielfeldrand gelegt und der betreffende Spieler erhält 3 Strafpunkte.
Zum Schluß noch ein Hinweis zur Eingabe der Richtung. Diese muß immer dreistellig eingetippt werden. Sollten Sie einmal nur eine oder zwei Ziffern eingeben wollen, füllen Sie die restlichen Stellen mit Leerzeichen aus.
Noch ein Tip für die, die etwas Probleme mit den Winkeln haben: Zeichnen Sie sich einfach auf ein Blatt Papier oder auf einen Bierdeckel die markantesten Winkel und halten die Gerade, die den Nullpunkt darstellt in die gleiche Richtung, wie der Strich im oberen Bildschirmrand zeigt. Dann kann man den Winkel auf dem Zettel ablesen.

Torsten Lukassen

HAMMURABI

Hammurabi ist ein Text-Adventure, bei dem Sie nicht wochenlang nach einem richtigen Weg oder Schlüssel suchen müssen. Sie können das Spiel auf jeden Fall zu Ende bringen. Der Ausgang hängt jedoch ganz von Ihrem Entscheidungsvermögen und Ihren Führungsfähigkeiten ab. Als Regierungsoberhaupt müssen Sie 20 Jahre lang ein Land regieren und dem Volk zu Wohlstand verhelfen. Sie können Land kaufen und bebauen, Söldner anwerben und Kriege führen, Korruptionen aufdecken oder auf Goldsuche gehen. Dies war nur ein kleiner Einblick in das große Repertoire Ihrer Möglichkeiten. Nach Ablauf der Regierungszeit entscheidet das Volk, ob Sie weiter regieren dürfen oder gestürzt werden.
Nach dem Start des Programmes können alle möglichen Entscheidungen aufgelistet werden. Während des Spiels ist dann nur noch die Kennziffer der von Ihnen getroffenen Entscheidung einzugeben. Alles Weitere erledigt der Computer im Dialog mit Ihnen.
Und nun viel Spaß und eine lange und weise Regierungszeit.

Markus Paul

```
10 REM **********************
11 REM *                    *
12 REM *  -- MINI-GOLF --   *
13 REM * COPYRIGHT 1984 BY  *
14 REM *   KLAUS VIETZKE    *
15 REM *                    *
16 REM *   BEARBEITET VOM   *
17 REM *      TEAM DES      *
18 REM *  AKTUELL VERLAGES  *
19 REM *      MUENCHEN      *
20 REM *                    *
21 REM * BEN!TIGTE GERITE:  *
22 REM * TI 99/4 A KONSOLE  *
23 REM *                    *
24 REM * SPEICHERBELEGUNG:  *
25 REM *    9062 BYTES      *
26 REM *                    *
27 REM ******************be**
100 CALL CLEAR
110 RANDOMIZE
120 DIM NA$(4),PU(4),SCH(4),
HX(7),HY(7)
130 CALL SCREEN(16)
140 PRINT "WOLLT IHR DIE SPI
ELREGELN": :"SEHEN ? >J< >N<
"
150 CALL KEY(0,K,S)
160 IF S=0 THEN 150
170 IF K=74 THEN 210
180 IF K=78 THEN 570
190 CALL SOUND(100,2000,0)
200 GOTO 150
210 CALL CLEAR
220 PRINT "....MINIGOLF":"..
..========": :"IHR SEID AUF
DEM MINIGOLF-": :"PLATZ.": :
230 PRINT "ES KOENNEN 1-4 PE
RSONNEN....MITSPIELEN.": :"M
AN KANN..UNTER 5 VER-......S
CHIEDENEN SCHWIERIGKEIT-"
240 PRINT "STUFEN WAEHLEN.IC
H EMPFEHLE AM ANFANG NR.1.":
 :"ALLE EINGABEN MUESSEN MIT
...>ENTER< ABGESCHLOSSEN":
250 PRINT "WERDEN.": :"BITTE
 EINE TASTE DRUECKEN.":
260 CALL KEY(0,K,S)
270 IF S=0 THEN 260
280 CALL CLEAR
290 PRINT "MAN MUSS IM SPIEL
 3 WERTE...EINGEBEN;": : :"1
. DIE ABSCHUSS-HOEHE": : :"2
. DEN ABSCHUSS-WINKEL": : :"
UND": :
300 PRINT :"3. DIE SCHUSS-ST
AERKE.": :"BITTE EINE TASTE
DRUECKEN."
310 CALL KEY(0,K,S)
320 IF S=0 THEN 310
330 CALL CLEAR
340 PRINT "DIE ABSCHUSS-HOEH
E GIBT MAN DURCH DIE TASTEN'
E','X' EIN.": :"DURCH'E' WIR
D DIE ABSCHUSSS-HOEHE NACH O
BEN,"
350 PRINT "UND DURCH'X'NACH
UNTEN VER -SCHOBEN.": :"UND"
: :"ALLE EINGABEN MIT   >ENT
ER< ABSCHLIESEN.": :
360 PRINT "EINE TASTE DRUECK
EN."
370 CALL KEY(0,K,S)
380 IF S=0 THEN 370
390 CALL CLEAR
400 PRINT "DER ABSCHUSS-WINK
EL WIRD SO EINGEGEBEN:":"'E'
 NACH OBEN.":"'R' NACH OBEN/
LINKS.":"'D' NACH RECHTS."
410 PRINT "'C' NACH UNTEN/RE
CHTS.":"'X' NACH UNTEN.":"'Z
' NACH UNTEN/RECHTS.":"'S' N
ACH LINKS.":"UND":
420 PRINT "'W'NACH OBEN/LINK
S.":"DER WINKEL WIRD ANGEZEI
GT.": :"EINE TASTE DRUECKEN.
"
430 CALL KEY(0,K,S)
440 IF S=0 THEN 430
450 CALL CLEAR
460 PRINT "DIE SCHUSS-STAERK
E(1-9) WIRDDURCH DRUECKEN DE
R TASTE MITDER ZIFFER EINGEG
EBEN.": :
470 PRINT "NACH SPIEL-ENDE(D
IE PUNKT-  ZAHL WIRD ANGEZEI
GT)":"GEHT ES DURCH DRUECKEN
 EINERTASTE WEITER.": :
480 PRINT "ALLES VERSTANDEN?
 >J< >N<"
490 CALL KEY(0,K,S)
500 IF S=0 THEN 490
510 IF K=78 THEN 210
520 IF K=74 THEN 550
530 CALL SOUND(100,2000,0)
540 GOTO 490
550 CALL CLEAR
560 REM ZEICHEN DEF.
570 A$="007E7E7E7E7E7E00"
580 FOR I=112 TO 116
590 CALL CHAR(I,A$)
600 NEXT I
610 CALL CHAR(96,"")
620 CALL CHAR(97,"183C7EFFFF
7E3C18")
630 CALL CHAR(104,"004A6A2A3
A32121")
640 CALL CHAR(105,"00086B2E3
C0C08")
650 CALL CHAR(106,"000008CB7
5353E08")
660 CALL CHAR(107,"008183C32
C242404")
```

```
670 CALL CHAR(108,"020222301
0105040")
680 CALL CHAR(109,"004062262
42C282")
690 CALL CHAR(110,"008989DD5
2763C38")
700 CALL CHAR(111,"")
710 CALL CHAR(120,"3C7EFFFFF
FFF7E3C")
720 CALL CHAR(128,"FFFFFFFFF
FFFFFFF")
730 CALL CHAR(129,"FFFFFFFFF
FFFFFFF")
740 REM ABFRAGE DER ANZAHL
    DER PERSONNEN UND DER
    GEWUENSCHTEN
    SCHWIERIGKEIT
750 CALL CLEAR
760 PRINT : :"WIEVIELE PERSO
NEN WOLLEN": :
770 INPUT "MITSPIELEN(1-4)":
A
780 IF (A<1)+(A>4)THEN 760
790 FOR I=1 TO A
800 CALL CLEAR
810 PRINT : :"DEN NAMEN DES
";I;".SPIELERS": :
820 INPUT "EINGEBEN:":NA$(I)
830 PRINT : :"GEWUENSCHTE SC
HWIERIGKEIT": :
840 INPUT "EINGEBEN(1-5)":SC
H(I)
850 IF (SCH(I)>5)+(SCH(I)<1)
THEN 830
860 NA$(I)=SEG$(NA$(I),1,8)
870 NEXT I
880 CALL CLEAR
890 CALL SCREEN(2)
900 FOR I=1 TO 8
910 CALL COLOR(I,2,3)
920 NEXT I
930 CALL COLOR(9,7,15)
940 CALL COLOR(10,13,3)
950 CALL COLOR(11,2,15)
960 CALL COLOR(12,16,15)
970 REM  SPIELFELDAUFBAU
980 PRINT : :"hijkmijjnnmmkk
hiiihnmohijknmnsppppppppppppp
pppppppppppshir''''''''''''
''''''''''rj"
990 PRINT "jr''''''''''''''
''''''''rhhr''''''''''''''
''''''''riir''''''''''''''
''''''''rj"
1000 PRINT "jr''''''''''''''
''''''''riir''''''''''''''
''''''''rklsppppppppppppppp
pppq''''''rj"
1010 PRINT "jSPIELERoo:ooooo
ooor''''''rklBALLHOEHE:ooooo
ooor''''''rhmWINKELooo:ooooo
ooor''''''rh"
1020 PRINT "jSTAERKEoo:ooooo
ooor''''''rhkVERSUCHoo:kljoo
ojkr''''''rhkPUNKTEooo:jmooo
okjr''''''rk"
1030 PRINT "isppppppppppppp
pppq''''''rklr''''''''''''''
''''''''''rjjr''''''''''''''
''''''''''ri"
1040 PRINT "mr''''''''''''''
''''''''''rnnr''''''''''''''
''''''''''rjjr''''''''''''''
''''''''''ri"
1050 PRINT "ir''''''''''''''
''''''''''roospppppppppppppp
pppppppppskloijnmklijmnkjlo
ijkhnmkjhjhi";
1060 REM SPIELROUTINE
1070 FOR SPIEL=1 TO A
1080 B=1
1090 C=6-SCH(SPIEL)
1100 H=0
1110 ST=1
1120 VER=0
1130 WI=1
1140 X=8
1150 Y=5
1160 REM POSITIONEN DER
        HINDERNISSE
1170 RESTORE 1180
1180 DATA 11,16,21,21,16,11
1190 FOR I=1 TO 6
1200 READ D
1210 HX(I)=INT(RND*4)+D
1220 NEXT I
1230 HX(7)=INT(RND*3)+6
1240 HY(1)=3
1250 HY(2)=6
1260 HY(3)=3
1270 HY(4)=20
1280 HY(5)=17
1290 HY(6)=20
1300 HY(7)=INT(RND*4)+18
1310 ON C GOTO 1320,1410,144
0,1470,1500
1320 FOR Q=9 TO 13 STEP 2
1330 CALL HCHAR(18,Q,97)
1340 NEXT Q
1350 FOR I=-1 TO 1
1360 FOR Q=-1 TO 1
1370 IF (Q=0)+(I=0)THEN 1390
1380 CALL HCHAR(HY(7)+Q,HX(7
)+I,97)
1390 NEXT Q
1400 NEXT I
1410 CALL HCHAR(10,24,97)
1420 CALL HCHAR(10,28,97)
1430 CALL HCHAR(12,25,97,2)
1440 FOR Q=4 TO 6
```

```
1450 CALL VCHAR(HY(Q),HX(Q),
128,3)
1460 NEXT Q
1470 FOR Q=1 TO 3
1480 CALL VCHAR(HY(Q),HX(Q),
128,3)
1490 NEXT Q

1500 CALL HCHAR(HY(7),HX(7),
129)
1510 REM ABFRAGE VON BALL-
     HOEHE,ABSCHUSSWINKEL
     UND SCHUSSSTAERKE
1520 N$=NA$(SPIEL)
1530 VV=13
1540 HH=10
1550 GOSUB 3140
1560 REM BALL HOEHEN-
        ABFRAGE
1570 N$="EINGEBEN"
1580 HH=11
1590 GOSUB 3140
1600 CALL HCHAR(Y,X,120)
1610 CALL KEY(0,K,S)
1620 IF S=0 THEN 1610
1630 IF K=13 THEN 1790
1640 IF K=69 THEN 1690
1650 IF K=88 THEN 1740
1660 GOTO 1610
1670 CALL HCHAR(Y,X,120)
1680 GOTO 1610
1690 CALL HCHAR(Y,X,96)
1700 Y=Y-1
1710 IF Y=2 THEN 1720 ELSE 1
670
1720 Y=3
1730 GOTO 1670
1740 CALL HCHAR(Y,X,96)
1750 Y=Y+1
1760 IF Y=9 THEN 1770 ELSE 1
670
1770 Y=8
1780 GOTO 1670
1790 CALL HCHAR(11,14,111,8)
1800 REM ABSCHUSSWINKEL-
        ABFRAGE
1810 HH=12
1820 VV=13
1830 N$="EINGEBEN"
1840 GOSUB 3140
1850 CALL KEY(0,K,S)
1860 IF S=0 THEN 1850
1870 IF K=13 THEN 2410
1880 CALL HCHAR(12,14,111,8)
1890 CALL HCHAR(12,16,47)
1900 IF K=68 THEN 2000
1910 IF K=67 THEN 2050
1920 IF K=88 THEN 2100
1930 IF K=90 THEN 2150
1940 IF K=83 THEN 2200
1950 IF K=87 THEN 2250
1960 IF K=69 THEN 2300
1970 IF K=82 THEN 2350
1980 CALL SOUND(100,2000,0)
1990 GOTO 1850
2000 CALL HCHAR(12,15,82)
2010 CALL HCHAR(12,17,82)
2020 B=1
2030 H=0

2040 GOTO 1850
2050 CALL HCHAR(12,15,82)
2060 CALL HCHAR(12,17,85)
2070 B=1
2080 H=1
2090 GOTO 1850
2100 CALL HCHAR(12,15,85)
2110 CALL HCHAR(12,17,85)
2120 B=0
2130 H=1
2140 GOTO 1850
2150 CALL HCHAR(12,15,76)
2160 CALL HCHAR(12,17,85)
2170 B=-1
2180 H=1
2190 GOTO 1850
2200 CALL HCHAR(12,15,76)
2210 CALL HCHAR(12,17,76)
2220 B=-1
2230 H=0
2240 GOTO 1850
2250 CALL HCHAR(12,15,76)
2260 CALL HCHAR(12,17,79)
2270 B=-1
2280 H=-1
2290 GOTO 1850
2300 CALL HCHAR(12,15,79)
2310 CALL HCHAR(12,17,79)
2320 B=0
2330 H=-1
2340 GOTO 1850
2350 CALL HCHAR(12,15,82)
2360 CALL HCHAR(12,17,79)
2370 B=1
2380 H=-1
2390 GOTO 1850
2400 REM ABSCHUSSSTAERKE-
        ABFRAGE
2410 HH=13
2420 GOSUB 3140
2430 CALL KEY(0,K,S)
2440 IF S=0 THEN 2430
2450 IF K=13 THEN 2520
2460 IF (K>48)*(K<58)THEN 24
70 ELSE 2430
2470 CALL HCHAR(13,14,111,8)
2480 CALL HCHAR(13,16,K)
2490 ST=K-48
2500 GOTO 2430
2510 REM SCHUSSROUTINE
2520 VER=VER+1
2530 N$=STR$(VER)
2540 HH=14
2550 VV=15
```

```
2560 GOSUB 3140
2570 FOR I=1 TO ST*9
2580 CALL HCHAR(Y,X,120)
2590 REM ABFRAGE OB DER
     BALL AUF EIN
     HINDERNIS STOESST.
2600 YY=Y+H
2610 XX=X+B
2620 CALL GCHAR(YY,XX,V)
2630 IF V<>96 THEN 2690
2640 CALL HCHAR(Y,X,96)
2650 CALL HCHAR(YY,XX,120)
2660 Y=YY
2670 X=XX
2680 GOTO 3110
2690 IF V=112 THEN 2750
2700 IF V=114 THEN 2720
2710 IF V=115 THEN 2780 ELSE
 2820
2720 B=-B
2730 CALL SOUND(100,1000,0)
2740 GOTO 2580
2750 H=-H
2760 CALL SOUND(100,1111,0)
2770 GOTO 2580
2780 B=-B
2790 H=-H
2800 CALL SOUND(100,1222,0)
2810 GOTO 2580
2820 IF V<>113 THEN 2890
2830 IF (B=1)+(B=0)THEN 2750
2840 IF H=0 THEN 2720
2850 IF (YY=9)*(H=1)THEN 278
0
2860 IF (YY=16)*(H=-1)THEN 2
780
2870 GOTO 2720
2880 REM TEFFER-ROUTINE
2890 IF V<>129 THEN 3020
2900 CALL HCHAR(Y,X,96)
2910 CALL HCHAR(YY,XX,120)
2920 FOR Q=110 TO 1100 STEP
55
2930 CALL SOUND(-100,Q,0,Q+1
10,4,Q+30,6)
2940 NEXT Q
2950 PU(SPIEL)=PU(SPIEL)+((5
0-SCH(SPIEL)*2)-VER)*SCH(SPI
EL)
2960 N$=STR$(PU(SPIEL))
2970 VV=14
2980 HH=15
2990 GOSUB 3140
3000 CALL KEY(0,K,S)
3010 IF S=0 THEN 3000 ELSE 3
190
3020 IF V<>128 THEN 3100
3030 IF B=0 THEN 2750
3040 IF (YY=5)+(YY=6)THEN 30
60 ELSE 3050
3050 IF (YY=19)+(YY=20)THEN
3070 ELSE 2720
3060 IF (XX=HX(1))+(XX=HX(3)
)THEN 3090 ELSE 3080
3070 IF XX=HX(5)THEN 3090
3080 IF H=1 THEN 2780 ELSE 2
720
3090 IF H=-1 THEN 2780 ELSE
2720
3100 IF V<>97 THEN 3110 ELSE
 2780
3110 NEXT I
3120 GOTO 1810
3130 REM DRUCK-ROUTINE
3140 FOR U=1 TO LEN(N$)
3150 CALL HCHAR(HH,VV+U,ASC(
SEG$(N$,U,1)))
3160 NEXT U
3170 RETURN
3180 REM LOESCHEN DER
        HINDERNISSE
3190 FOR U=1 TO 6
3200 CALL VCHAR(HY(U),HX(U),
96,3)
3210 NEXT U
3220 FOR U=10 TO 15
3230 CALL HCHAR(U,14,111,8)
3240 CALL HCHAR(U,24,96,5)
3250 NEXT U
3260 FOR U=17 TO 22
3270 CALL HCHAR(U,5,96,9)
3280 NEXT U
3290 NEXT SPIEL
3300 REM AUSGABE DER
        PUNKTE
3310 CALL VCHAR(1,1,32,768)
3320 CALL SCREEN(16)
3330 FOR I=1 TO 8
3340 CALL COLOR(I,2,16)
3350 NEXT I
3360 PRINT : :" PUNKTE:   NA
ME:": : :
3370 FOR I=1 TO A
3380 PRINT : :" ";PU(I);"
  ";NA$(I): :
3390 NEXT I
3400 PRINT "EINE TASTE DRUEC
KEN!"
3410 CALL KEY(0,K,S)
3420 IF S=0 THEN 3410
3430 CALL CLEAR
3440 REM ABFRAGE OB DER
     SPIELER WEITERSPIELEN
    ODER AUFHOEREN MOECHTE.
3450 PRINT : :"BITTE WAEHLEN
;": :" 'W' FUER WEITERSPIELE
N": :" 'A' FUER AUFHOEREN":
:" UND": :
3460 PRINT " 'N' FUER NEU-AN
```

```
FANGEN."
3470 CALL KEY(0,K,S)
3480 IF S=0 THEN 3470
3490 IF K=87 THEN 880
3500 IF K=65 THEN 3540
3510 IF K=78 THEN 3560
3520 CALL SOUND(100,2000,0)
3530 GOTO 3470
3540 CALL CLEAR
3550 END

3560 FOR I=1 TO A
3570 PU(I)=0
3580 NEXT I
3590 GOTO 100
3600 END
```

```
100 REM*********************
110 REM*                    *
120 REM*    HAMMURABI       *
130 REM*  Copyright  by     *
140 REM*   Markus Paul      *
150 REM*                    *
160 REM* Bearbeitet  vom    *
170 REM*    Team  des       *
180 REM* Aktuell Verlages   *
190 REM*     Muenchen       *
200 REM*                    *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole   *
230 REM*                    *
240 REM* Speicherbelegung   *
250 REM*    7415 Bytes      *
260 REM*********************
270 REM
280 RANDOMIZE
290 REM VARIABLEN WERDEN ZUF
AELLIG BESTIMMT
300 E1=INT(RND*999)+1
310 IF E1<250 THEN 300
320 E2=INT(E1/3)
330 E3=E2
340 E4=INT(RND*699)+1
350 Z1=500000
360 Z2=5000
370 Z3=2000
380 CALL CLEAR
390 REM FRAGE NACH ANLEITUNG

400 PRINT TAB(11);"HAMMURABI
"
410 PRINT : :"BENOETIGEN SIE
 EINE":"ANLEITUNG ? <J><N>"
420 CALL KEY(0,A,B)
430 IF B=0 THEN 420
440 IF (A=74)+(A=106)THEN 25
40
450 REM  SPIELBEGINN(GRUNDWE
RTE AUSGEBEN)
460 FOR J=1 TO 20
470 A2=INT(RND*99)+1
480 A3=INT(RND*99)+1
490 PRINT : : : :
500 PRINT "JAHR ";J
510 PRINT "EINWOHNER :";E1
520 PRINT "MAENNER :";E2
530 PRINT "SOLDATEN :";S
540 PRINT "FARMER :";F
550 PRINT "SKLAVEN :";E4
560 IF A10=-1 THEN 2850
570 IF A9>0 THEN 2900
580 DISPLAY : :"GETREIDE :";
Z1;"T"
590 DISPLAY :"LAND :";Z2;"HA
"
600 DISPLAY "LANDPREIS :";A3
;"T"
610 DISPLAY :"GOLD :";Z3;"T"
```

```
620 DISPLAY "GOLDPREIS :";A2
;"T"
630 DISPLAY : : : :
640 PRINT
650 REM ABFRAGE WAS TUN
660 INPUT "HAMMURABI,WAS NUN
? ":TUN$
670 IF TUN$="1" THEN 820
680 IF TUN$="2" THEN 910
690 IF TUN$="3" THEN 1000
700 IF TUN$="4" THEN 1090
710 IF TUN$="5" THEN 1180
720 IF TUN$="6" THEN 1300
730 IF TUN$="7" THEN 1390
740 IF TUN$="8" THEN 1520
750 IF TUN$="9" THEN 1600
760 IF TUN$="10" THEN 1680
770 IF TUN$="11" THEN 1750
780 IF TUN$="12" THEN 1870
790 IF TUN$="13" THEN 2010
800 IF TUN$="" THEN 2350
810 IF TUN$>"13" THEN 640
820 REM LAND KAUFEN
830 PRINT
840 INPUT "WIEVIEL HA WILLST
 DU ?":A4
850 IF A4*A3>Z1 THEN 890
860 Z2=Z2+A4
870 Z1=Z1-(A3*A4)
880 GOTO 580
890 PRINT "DAS KANNST DU NIC
HT BEZAHLEN"
900 GOTO 580
910 REM LAND VERKAUFEN
920 PRINT
930 INPUT "WIEVIEL HA WILLST
 DU VER-...KAUFEN ?":A5
940 IF A5>Z2 THEN 980
950 Z2=Z2-A5
960 Z1=Z1+(A3*A5)
970 GOTO 580
980 PRINT "DU HAST NUR";Z2;"
HA LAND"
990 GOTO 580
1000 REM  GOLD KAUFEN
1010 PRINT
1020 INPUT "WIEVIEL T WILLST
 DU ?":A6
1030 IF A6*A2>Z1 THEN 1070
1040 Z3=Z3+A6
1050 Z1=Z1-(A2*A6)
1060 GOTO 580
1070 PRINT "DAS KANNST DU NI
CHT BEZAHLEN"
1080 GOTO 580
1090 REM GOLD VERKAUFEN
1100 PRINT
1110 INPUT "WIEVIEL T WILLST
 DU VERKAUF-EN ?":A7
1120 IF A7>Z3 THEN 1160
1130 Z3=Z3-A7
1140 Z1=Z1+(A2*A7)
1150 GOTO 580
1160 PRINT "DU HAST NUR";Z3;
"T GOLD"
1170 GOTO 580
1180 REM SOLDATEN ANHEUERN
1190 A1=INT(RND*99)+1
1200 PRINT
1210 PRINT "DER WEHRSOLD BET
RAEGT";A1;"T"
1220 INPUT "WIEVIEL SOLDATEN
 WILLST DU ?":SN
1230 IF SN>E3 THEN 1280
1240 S=S+SN
1250 Z1=Z1-(A1*SN)
1260 E3=E3-SN
1270 GOTO 580
1280 PRINT "ES STEHEN DIR NU
R";E3;"MAENNER ZUR VERFUEGUN
G"
1290 GOTO 580
1300 REM SOLDATEN FEUERN
1310 PRINT
1320 INPUT "WIEVIELE SOLDATE
N WILLST DU ENTLASSEN ? ":SW
1330 IF SW>S THEN 1370
1340 S=S-SW
1350 E3=E3+SW
1360 GOTO 580
1370 PRINT "DU HAST NUR";S;"
SOLDATEN"
1380 GOTO 580
1390 REM BEVOELKERUNG ZAEHLE
N
1400 PRINT
1410 INPUT "EINE VOLKSZAEHLU
NG KOSTET...33752 T.OKAY ? (
J/N)":OK$
1420 IF ((OK$="J")+(OK$="j")
+(OK$="JA")+(OK$="ja"))=0 TH
EN 640
1430 Z1=Z1-33752
1440 PRINT : :
1450 E5=INT(E1/9)
1460 E7=INT(E1/9*3)
1470 E6=INT(E1/9*2)
1480 E8=INT(RND*49)+1
1490 PRINT " DEIN VOLK BESTE
HT AUS :"
1500 PRINT E2;"MAENNERN":E5;
"RENTNERN":E7;"FRAUEN":E6;"K
INDERN":E8;"RATTEN"
1510 GOTO 580
1520 REM RATTEN TOETEN
1530 PRINT
1540 INPUT "EIN RATTENFAENGE
R KOSTET....33742 T.OKAY ? (
J/N)":OK$
1550 IF ((OK$="J")+(OK$="JA"
)+(OK$="j")+(OK$="ja"))=0 TH
EN 640
```

```
1560 Z1=Z1-33742
1570 PRINT E8;"RATTEN GETOET
ET."
1580 E8=0
1590 GOTO 580
1600 REM KORRUPTION UNTERSUC
HEN
1610 PRINT
1620 INPUT "EINE UNTERSUCHUN
G DER KOR-..RUPTION KOSTET 2
0000 T.OKAY ? (J/N)":OK$
1630 IF ((OK$="J")+(OK$="JA"
)+(OK$="j")+(OK$="ja"))=0 TH
EN 640
1640 Z1=Z1-20000
1650 A8=INT(RND*9)+1
1660 PRINT A8;"POLITIKER GEF
EUERT"
1670 GOTO 580
1680 REM LAND BEBAUEN
1690 PRINT
1700 INPUT "WIEVIEL HA SOLLE
N BEBAUT....WERDEN ? ":A9
1710 IF A9>Z2 THEN 1730
1720 GOTO 580
1730 PRINT "DU HAST NUR";Z2;
"HA"
1740 GOTO 580
1750 REM FARMER ANHEUERN
1760 PRINT
1770 FAN=INT(RND*99)+1
1780 PRINT "DER LOHN FUER EI
NEN FARMER  BETRAEGT";FAN;"T
."
1790 INPUT "WIEVIELE ?":FN
1800 IF FN>E3 THEN 1850
1810 Z1=Z1-(FAN*FN)
1820 F=F+FN
1830 E3=E3-FN
1840 GOTO 580
1850 PRINT "DU HAST NUR";E3;
"MAENNER ZUR VERFUEGUNG"
1860 GOTO 580
1870 REM GOLD SUCHEN
1880 PRINT
1890 IF A10=-1 THEN 2960
1900 INPUT "EINE EXPEDITION
KOSTET 500 TGOLD.OKAY ? (J/N
)":OK$
1910 IF ((OK$="J")+(OK$="JA"
)+(OK$="j")+(OK$="ja"))=0 TH
EN 640
1920 A10=INT(RND*2)-1
1930 IF A10=-1 THEN 1970
1940 PRINT "DU HAST 500 T GO
LD ZUM FEN- STER HINAUSGESCH
MISSEN."
1950 Z3=Z3-500
1960 GOTO 580
1970 PRINT "**********
*****"
1980 PRINT "DU HAST GOLD GEF
UNDEN"
1990 PRINT "**********
*****"
2000 GOTO 580
2010 REM KRIEG FUEHREN
2020 K1=INT(RND*9)+1
2030 IF S=0 THEN 3160
2040 PRINT
2050 PRINT "DIE FOLGEN DES K
RIEGES :"
2060 K2=INT(RND*9999)+1
2070 K3=INT(RND*9999)+1
2080 K4=INT(RND*9999)+1
2090 K5=INT(RND*999)+1
2100 IF K1=1 THEN 2230
2110 PRINT K2;"T MEHR GETREI
DE":K3;"HA MEHR LAND":K4;"T
MEHR GOLD":K5;"MEHR SKLAVEN"
2120 Z1=Z1+K2
2130 Z2=Z2+K3
2140 Z3=Z3+K4
2150 E4=E4+K5
2160 K6=INT(RND*4)+1
2170 PRINT K6;"MANN VERLUSTE
 IM HEER"
2180 S=S-K6
2190 E2=E2-K6
2200 E1=E1-K6
2210 PRINT
2220 GOTO 2320
2230 PRINT
2240 PRINT K2;"T WENIGER GET
REIDE":K3;"HA WENIGER LAND":
K4;"T WENIGER GOLD"
2250 Z1=Z1-K2
2260 IF Z1<0 THEN 3100
2270 Z2=Z2-K3
2280 IF Z2<0 THEN 3120
2290 Z3=Z3-K4
2300 IF Z3<0 THEN 3140
2310 PRINT
2320 PRINT "DU HAST JETZT:"
2330 PRINT Z1;"GETREIDE":Z2;
"LAND":Z3;"GOLD"
2340 GOTO 640
2350 REM TUN=""
2360 PRINT
2370 PRINT "WIEVIEL DER";Z1;
"T GETREIDE SOLLEN FUER BROT
":"VERBRAUCHT WERDEN ?"
2380 INPUT B1
2390 IF B1<INT(E1*20)THEN 24
50 ELSE 2780
2400 Z1=Z1-B1
2410 IF J=20 THEN 2990
2420 NEXT J
```

```
2430 GOTO 2990
2440 REM ZUWENIG BROT
2450 G=INT(RND*99)+1
2460 PRINT "DAS WAR ZU WENIG
.ES SIND";G;"MENSCHEN VERHUN
GERT."
2470 GN=GN+G
2480 Z1=Z1-B1
2490 E1=E1-G
2500 E2=E2-INT(G/3)
2510 E3=E2
2520 IF J=19 THEN 2990
2530 GOTO 2420
2540 REM ANLEITUNG
2550 CALL CLEAR
2560 PRINT
2570 PRINT TAB(11);"ANLEITUN
G"
2580 PRINT
2590 PRINT "SIE SOLLEN ALS H
AMMURABI,DERHERRSCHER VON SU
MMARIA,DAS..LAND 20 JAHRE LA
ND REGIEREN."
2600 PRINT "HAUPTZAHLUNGSMIT
TEL IST GE- TREIDE."
2610 PRINT "DABEI KOENNEN SI
E FOLGENDE..BEFEHLE ANWENDEN
 :":"TASTE DRUECKEN"
2620 CALL KEY(0,A,B)
2630 IF B THEN 2650
2640 GOTO 2620
2650 CALL CLEAR
2660 PRINT "1. LAND KAUFEN":
"2. LAND VERKAUFEN":"3. GOLD
 KAUFEN"
2670 PRINT "4. GOLD VERKAUFE
N":"5. SOLDATEN ANHEUERN":"6
. SOLDATEN FEUERN"
2680 PRINT "7. BEVOELKERUNG
ZAEHLEN":"8. RATTEN TOETEN":
"9. KORRUPTION UNTERSUCHEN":
"10. LAND BEBAUEN"
2690 PRINT "11. FARMER ANHEU
ERN":"12. GOLD SUCHEN":"13.
KRIEG FUEHREN"
2700 PRINT "WENDEN SIE DIESE
 BEFEHLE....MOEGLICHST ERFOL
GREICH AN...EIN JAHR BEENDEN
 SIE DURCH..EINGEBEN VON"
2710 PRINT "ENTER."
2720 PRINT "DRUECKEN SIE EIN
E TASTE"
2730 CALL KEY(0,A,B)
2740 IF B THEN 460
2750 GOTO 2730
2760 END
2770 REM X PERSONEN ZUGEZOGE
N
2780 E9=INT(RND*199)+1
2810 E2=E2+INT(E9/3)
2820 E3=E3+INT(E9/3)
2830 GOTO 2420
2840 REM DIE MINE BRACHTE EI
N
2850 A11=INT(RND*99)+1
2860 PRINT "DIE GOLDMINE BRA
CHTE";A11;"T GOLD"
2870 Z3=Z3+A11
2880 GOTO 570
2890 REM DAS BEBAUTE LAND BR
ACHTE EIN
2900 A12=INT(RND*9)+1
2910 PRINT "DAS BEBAUTE LAND
 BRACHTE";A12;"T/HA EIN"
2920 Z1=Z1+(A9*A12)
2930 A9=0
2940 GOTO 580
2950 REM DU BRAUCHST KEINE M
INE MEHR
2960 PRINT "DU HAST SCHON EI
NE MINE"
2970 GOTO 580
2980 REM ENDE 1
2990 PRINT
3000 IF GN=0 THEN 3070
3010 PRINT "INSGESAMT SIND";
GN;"VERHUNGERT"
3020 PRINT "DAS WAR SCHON GA
NZ GUT,JE-..DOCH DAS VOLK LI
EBTE SIE....NICHT.SIE SIND G
ELYNCHT.....WORDEN."
3030 INPUT "WOLLEN SIE NOCHM
AL REGIEREN?":JA$
3040 IF ((JA$="J")+(JA$="JA"
)+(JA$="j")+(JA$="ja"))<>0 T
HEN 280
3050 END
3060 REM ENDE 2
3070 INPUT "DA KEINER VERHUN
GERT IST....BITTET SIE DAS V
OLK WEITER..ZU REGIEREN.NEHM
EN SIE DAS..ANGEBOT AN?":JA$
3080 GOTO 3040
3090 REM NULLSETZUNG EINIGER
 VARIABLEN
3100 Z1=0
3110 GOTO 2270
3120 Z2=0
3130 GOTO 2290
3140 Z3=0
3150 GOTO 2310
3160 REM KEINE SOLDATEN DA
3170 PRINT "DU HAST NOCH KEI
NE SOLDATEN"
3180 GOTO 660
2790 PRINT E9;"PERSONEN ZUGE
ZOGEN"
2800 E1=E1+E9
```

PHYSIK

Das Programm dient der schnellen Suche nach Daten im Periodensystem der Elemente. Analysendaten, die im Laborversuch oder auch teilweise rechnerisch gewonnen wurden, können mit diesem Programm schnell eingeordnet werden.
Dabei wird jedes Element im Periodensystem fünfzehn verschiedenen Kriterien unterworfen. Insgesamt stehen also 1685 Daten zur Verfügung. Neben einer kurzen Bedienungsanleitung im Programm soll hier nun eine nähere Erläuterung der Bedienung hilfreich sein.
Wird ein Element mit seinem Namen oder Symbol aufgerufen, so werden seine 15 Kriterien auf dem Bildschirm sichtbar.

SYMBOL = smbolische Abkürzung des Elements
ATOMG. = Atomgewicht des Elements
OX-ZAHL = Oxidationszahl (bei mehreren erfolgt folgende Schreibweise: +– 2/4/6 d.h. die Oxidationszahlen können +– 2,4 und 6 sein.
AGGREGAT = Festlegung des Aggregatzustandes bei Raumtemperatur
S = fest
G = gasförmig
L = flüssig
SY = wird synthetisch hergestellt
SDP. = Siedepunkt
SMP. = Schmelzpunkt
DICHTE = Dichte
KOV-RADIUS = Kovalent Radius eines Atoms
A-RADIUS = Atomradius
I-RADIUS = Ionenradius (der Wert in Klammern z.B. 0,70 (+5) bedeutet, daß die 0,70 Angström bei der Ox-Zahl (+5) d. Elemente vorliegen.
A-VOLUMEN = Atombolumen
I-ENERGIE = Erste Ionisierungsenergie
EN = Elektronegativität
KRISTALL = Kristallstruktur des Elements
kubisch flächenzentriert
kubisch raumzentriert
Diamant
Kubisch
Hexagonal
Rhombisch
Tetragonal
Orthorhombisch
Monoklin

S/B – CHAR. = Säure und Base-Eigenschaften
S = Säure
S(+) = starke Säure
B = Base
B(+) = starke Base
S&B = amphotere Eigenschaften (sowohl Säure als auch Base)

Sollten sich durch neue Veröffentlichungen einzelne Kriterienwerte der Elemente leicht verändern, so können diese schnell in den Data Werten des ausgewählten Elements neu eingefügt werden. So bleibt es dem Benutzer z.B. selbst überlassen, die in diesem Pogramm gebräuchlichen Sdp. bzw. Smp.-Werte in Grad Celsius zu belassen oder sie in Kelvin umzurechnen.
Bei der Ausgabe der Elektronenkonfiguration werden die Hauptorbitale K-Q und die Nebenorbitale s,p,d,f aufgeführt. Der angezeigte Zahlenw ert entspricht der Anzahl an Elektronen in dem bestimmten Orbital.
Bei der Datenanalyse werden nur die Datenwerte berücksichtigt, die für eine Analyse bei Elementen von Nutzen sind. Folgende Suchkriterien stehen dabei zur Verfügung:

Atomgewicht
Siedepunkt
Schmelzpunkt
Dichte
Kovalentradius
Atomradius
Ionenradius
Atomvolumen
Ionisierungsenergie
Elektronegativität

Die Eingabe deszu untersuchenden Datenwertes kann zusätzlich durch eine Toleranz variiert werden. Beträgt die Toleranz z.B. 0,5 und wird ein Datenwert beim Atomgewicht von 91,22 eingegeben, so werden diejenigen Elemente ausgesucht und angezeigt, die eine Atommasse zwischen 90, 72 und 91,72 besitzen. Bei einer Toleranz von 0 wird dementsprechend nur das Element mit einer Masse von 91,22 angezeigt.
Die hier aufgeführten Programmdurchführungen können wahlweise auch auf einem EPSON Drucker z.B. FX 80 ausgedruckt werden. Vor dem Laden des Programmes von der Diskette *muß* die Eingabe
Call Files (1) ENTER
New ENTER erfolgen.
Bei der Programmdurchführung besteht zusätzlich noch die Möglichkeit, die gesamten Elemente des Periodensystems als Tafeln auf dem Bildschirm anzuzeigen.
Das Programm kann auf Diskette von mir bezogen werden (Andreas Pack, Eicheler Str. 60, 4690 Herne 2).

```
10 ! EINGABE ELEKTRONENKONFIGURATION
20 !
30 DIM T$(22):: FF$="DSK1."
40 CALL CLEAR :: DISPLAY AT(1,5):"SYMBOL DES ELEMENTS ?" :: ACCEPT AT(3,1):A$ ::
 DISPLAY AT(1,1):A$
50 X=3 :: FOR I=1 TO 22 :: DISPLAY AT(X,1):"EINGABE D.";I;"-TEN STELLE" :: ACCEP
T AT(X,27):T$(I):: X=X+1 :: NEXT I
60 OPEN #1:FF$&A$,SEQUENTIAL,INTERNAL,OUTPUT,VARIABLE 22
70 FOR I=1 TO 22
80 PRINT #1:T$(I)
90 NEXT I :: CLOSE #1
100 CALL CLEAR :: DISPLAY AT(15,1):"WEITERE EINGABE ?J/N"
110 CALL KEY(0,K,S):: IF S=0 THEN 110 :: IF K=74 THEN 40
120 END
```

```
2 !  ************************
*
3 !  *    PERIODENSYSTEM
*
4 !  *        DER
*
5 !  *          ELEMENTE
*
6 !  *
*
7 !  * (C) ANDREAS PACK
*
8 !  *      4690 HERNE 2
*
10 ! *
*
11 ! *      TI EX-BASIC
*
12 ! *      ERWEITERUNG
*
13 ! *      DISKETTE
*
14 ! *      DRUCKER OPT.
*
15 ! *
*
16 ! *      24322 BYTES
*
17 ! *
*
18 ! ************************
*
19 CALL COLOR(1,1,5)
20 CALL CLEAR :: DISPLAY AT(
7,1):"P E R I O D E N S Y S
T E M":TAB(20);"";TAB(12);"D
 E R":TAB(20);"";TAB(7);"E L
 E M E N T E"
29 DISPLAY AT(20,2):"GEBRAUC
HSANWEISUNG? (J/N)"
30 ACCEPT AT(23,24)VALIDATE(
"J,N")BEEP SIZE(-1):X$ :: IF
 X$="J" THEN 34 ELSE 100
34 CALL COLOR(1,16,5,2,16,5,
3,16,5,4,16,5,5,16,5,6,16,5,
7,16,5,8,16,5,9,16,5,10,16,5
,11,16,5)
35 DISPLAY AT(4,4)ERASE ALL:
"DIESES PROGRAMM KANN IHNEN
 1648 VERSCHIEDENE DATEN AUS
 DEM PERIODENSYSTEM DER ELE-
 MENTE VERMITTELN."
37 DISPLAY AT(9,4):"DER PROG
RAMMABLAUF WIRD      DURCH ME
NUES GESTEUERT.AUF  GRUND DE
R GROSSEN DATENMENGEIST ES N
UR MOEGLICH DATEN"
39 DISPLAY AT(14,4):"BLOCKWE
ISE AUS DEM SPEICHER EINZULE
SEN. DIE BEIDEN      BLOECKE
 BESTEHEN EINMAL AUS DEN HAU
PT-UND NEBENGRUPPEN"
40 DISPLAY AT(19,4):"UND DEN
 LANTHANIDEN-UND      ACTINID
EN." :: PRINT "     MIT LEERT
ASTE WEITER"
41 CALL KEY(0,K,S):: IF S=0
THEN 41 :: IF K>32 THEN 41
50 CALL CLEAR :: DISPLAY AT(
3,3):"SIE KOENNEN IHRE EINZE
LNEN  ANALYSENDATEN IN DEN
      COMPUTER EINGEBEN UND
ERHAL-TEN NACH DER AUSWERTUN
G"
55 DISPLAY AT(8,1):"DIE LIST
E DER ELEMENTE DIE  IHREN DA
TEN ENTSPRECHEN!"
60 DISPLAY AT(11,4):"DABEI W
ERDEN SIE NACH DER   TOLERAN
ZBREITE GEFRAGT!(Bsp.TOLERAN
Z DER DICHTE =2, EIN-GABEWER
T =14.2 )"
65 DISPLAY AT(18,1):"ERGEBNI
S: ALLE ELEMENTE MIT DER DIC
HTE VON 12.2-16.2    WERDEN
AUSGEDRUCKT!" :: PRINT "
MIT LEERTASTE WEITER"
66 CALL KEY(0,K,S):: IF S=0
THEN 66 :: IF K>32 THEN 66
68 CALL CLEAR :: DISPLAY AT(
7,3):"BEIM ABRUF DER GESAMTD
ATEN  EINES ELEMENTES HABEN
SIE   DIE MOEGLICHKEIT, DIE
ELEK- TRONENSTRUKTUR DES ELE
MENTES"
70 DISPLAY AT(12,1):"AUSGEBE
N ZU LASSEN. DURCH   DIE ENO
RME DATENMENGE KANN  DIE ELE
KTRONENKONFIGURATION NUR VON
 DER DISKETTE        ABGERUF
EN WERDEN!"
72 PRINT "    MIT LEERTASTE
WEITER"
74 CALL KEY(0,K,S):: IF S=0
THEN 74 :: IF K>32 THEN 74
100 DIM B$(16),A$(75,16),S$(
7),T$(22),DI$(8)
105 CALL CHAR(96,"8080808080
808080",39,"000000FF"):: RES
TORE 2250 :: FOR A=1 TO 7 ::
 READ S$(A):: NEXT A
115 !E- LESEN
116 !
117 !
```

```
120 RESTORE :: FOR I=1 TO 16
 :: READ B$(I):: NEXT I
121 !MENUE
122 !
123 !
124 CALL CHAR(64,"0000004428
101010")
125 CALL CLEAR :: CALL COLOR
(1,16,5,2,16,5,3,16,5,4,16,5
,5,16,5,6,16,5,7,16,5,8,16,5
)
126 CALL COLOR(9,16,5,10,16,
5,11,16,5):: CALL SCREEN(9)
130 DISPLAY AT(5,4)BEEP:"1.
P.S.E.-(TAFELN)",,TAB(4);"2.
 GESAMTDATEN/ELEMENTE",TAB(4
);"   HAUPT-NEBENGRUPPEN"
140 DISPLAY AT(10,4):"3. GES
AMTDATEN/ELEMENTE",TAB(4);"
  LANTHANIDE/ACTINIDE",,TAB(
4);"4. DATENSUCHE/ANALYSE",,
TAB(4);"5. PROGRAMMENDE"
145 ON WARNING NEXT
150 DISPLAY AT(20,4):"EINGAB
E-NR.=> " :: ACCEPT AT(20,19
)VALIDATE("1,2,3,4,5")SIZE(-
1):X
160 ON X GOTO 170,1050,1080,
780,2260
170 CALL CLEAR :: CALL SCREE
N(14)
180 DISPLAY AT(5,4)BEEP:"1.
HAUPTGRUPPEN",,TAB(4);"2. NE
BENGRUPPEN",,TAB(4);"3. LANT
HANIDE",,TAB(4);"4. ACTINIDE
",,TAB(4);"5. RUECKSPRUNG"
190 DISPLAY AT(24,4):"EINGAB
E-NR.=> " :: X,Y,Z=0 :: ACCE
PT AT(24,19)VALIDATE("1,2,3,
4,5")SIZE(-1):X :: ON WARNIN
G NEXT
200 ON X GOTO 350,450,510,56
0,125
210 CALL CLEAR :: DISPLAY AT
(10,1):"MOECHTEN SIE NAME O.
 SYMBOL",,"DES ZU SUCHENDEN
ELEMENTS",,"EINGEBEN ? (N/S)
 :"
220 ACCEPT AT(14,20)SIZE(-1)
BEEP VALIDATE("N,S"):NAM$
230 IF NAM$<>"N" THEN 330
240 DISPLAY ERASE ALL AT(1,2
):"NAME :" :: ACCEPT AT(1,8)
:D$
250 FOR I=1 TO 75 :: IF D$<>
A$(I,1)THEN 260 ELSE 280
260 NEXT I :: X=1
270 PRINT :"   DATEN NICHT G
EFUNDEN " :: FOR Z=1 TO 500
:: NEXT Z :: ON X GOTO 230,3
30
275 !AUSWERTUNG
276 !
277 !
280 FOR J=1 TO 16 :: DISPLAY
 AT(J+3,1):B$(J);TAB(13);A$(
I,J):: NEXT J
290 DISPLAY AT(24,5):"AUSDRU
CKEN ? (J/N)"
300 CALL KEY(0,K,S):: IF S=0
 THEN 300 :: IF K=74 THEN GO
TO 2400
310 DISPLAY ERASE ALL AT(5,1
)BEEP:"E-STRUKTUR        (1)"
,TAB(5):"NEUE GESAMTDATEN (2
)",TAB(5):"RUECKSPRUNG
(3)":TAB(5):"EINGABE-NR.=>"
:: ACCEPT AT(11,19)VALIDATE(
"1,2,3")SIZE(-1):T
320 ON T GOTO 615,210,125
330 DISPLAY ERASE ALL AT(1,2
):"SYMBOL :" :: ACCEPT AT(1,
10):E$ :: FOR I=1 TO 75 :: I
F A$(I,2)<>E$ THEN 340 ELSE
280
340 NEXT I :: X=2 :: GOTO 27
0
345 !P-TAFEL
346 !
347 !
350 CALL COLOR(1,2,1):: U$="
HAUPTGRUPPEN" :: F=15 :: GOS
UB 1105
360 DISPLAY AT(9,2):"1A 2A
  3A 4A 5A 6A 7A 8A" :: DISP
LAY AT(13,2):"Li Be    B  C"
;TAB(27);"e"
370 CALL SPRITE(#1,72,9,81,2
5,#2,101,9,81,225,#3,72,9,81
,217,#4,78,9,97,145,#5,79,9,
97,169)
380 CALL SPRITE(#6,70,9,97,1
93,#7,78,9,97,217,#8,101,9,9
7,225,#9,67,9,113,193,#10,10
8,9,113,201)
390 CALL SPRITE(#11,65,9,113
,217,#12,114,9,113,225,#13,7
5,9,129,217,#14,114,9,129,22
5,#15,88,9,145,217)
400 CALL SPRITE(#16,101,9,14
5,225,#17,82,9,161,217,#18,1
10,9,161,225)
```

```
410 DISPLAY AT(15,2):"Na Mg
   Al Si P  S",TAB(27):" K
Ca    Ga Ge As Se Br"
420 DISPLAY AT(19,2):"Rb Sr
   In Sn Sb Te I"
430 DISPLAY AT(21,1):" Cs Ba
    Tl Pb Bi Po At",TAB(27):
" Fr Ra" :: X,Y,Z=0 :: GOSUB
 1100
440 CALL KEY(0,K,S):: IF S=0
 THEN 440 :: CALL DELSPRITE(
ALL):: GOTO 170
450 CALL COLOR(1,2,1):: U$="
NEBENGRUPPEN" :: X=0 :: Y=7
:: Z=3 :: F=6 :: GOSUB 1105
460 DISPLAY AT(7,1):" 3B 4B
5B 6B 7B   8B";TAB(20)," Sc
Ti V  Cr Mn Fe Co Ni":TAB(20
)," Y  Zr Nb Mo Tc Ru Rh Pd
"
470 DISPLAY AT(13,1):" La Hf
 Ta W  Re Os Ir Pt":TAB(27),
" Ac"
480 DISPLAY AT(18,1):" 1B 2B
":TAB(20),"öCuöZnö",TAB(20),
"ö  ö  ö",TAB(20),"öAgöCdö",
TAB(20),"ö  ö  ö",TAB(20),"ö
AuöHgö"
490 GOSUB 1100
500 CALL KEY(0,K,S):: IF S=0
 THEN 500 ELSE 170
510 CALL COLOR(1,2,1):: U$="
LANTHANIDE" :: X=2 :: Y,F=9
:: GOSUB 1105
520 DISPLAY AT(9,1):" 58 59
60 61 62 63 64 65 66":TAB(20
)," Ce Pr Nd Pm Sm Eu Gd Tb
D";TAB(27);CHR$(64)
530 GOSUB 1100
540 DISPLAY AT(17,1):" 67 68
 69 70 71":TAB(20)," Ho Er T
m Yb Lu" :: X=10 :: Y=1 :: Z
=12 :: GOSUB 1100
550 CALL KEY(0,K,S):: IF S=0
 THEN 550 ELSE 170
560 CALL COLOR(1,2,1):: U$="
ACTINIDE" :: X=0 :: Y,F=1i :
: GOSUB 1105
570 DISPLAY AT(7,1):" 90 91
92 93 94 95 96 97 98";TAB(20
)," Th Pa U  Np Pu Am Cm Bk
Cf" :: GOSUB 1100
580 DISPLAY AT(14,1):" 99  1
00 101 102 103 104 105":TAB(
20)," Es  Fm  Md  No  Lr  Un
q Unp"
590 FOR A=16 TO 19 :: FOR B=
3 TO 32 STEP 4 :: CALL HCHAR
(A,B,124):: NEXT B :: NEXT A
600 DISPLAY AT(21,1):" 106 1
07":TAB(27),"öUnhöUnsö",TAB(
27),"ö   ö   ö"
610 CALL KEY(0,K,S):: IF S=0
 THEN 610 ELSE 170
615 DISPLAY AT(20,5)BEEP:"MI
T DRUCKER ? (J/N)"
617 CALL KEY(0,K,S):: IF S=0
 THEN 617 :: IF K=74 THEN DE
$="P"
620 CALL CLEAR :: FOR A=3 TO
 9 STEP 3 :: FOR B=1 TO 21 :
: CALL VCHAR(B,A,96):: NEXT
B :: NEXT A :: FOR A=13 TO 1
7 STEP 4 :: FOR B=1 TO 21 ::
 CALL VCHAR(B,A,96):: NEXT B
 :: NEXT A
625 FOR C=3 TO 30 :: FOR D=1
 TO 24 STEP 3 :: CALL HCHAR(
D,C,39):: NEXT D :: NEXT C :
: X=1
627 FOR A=3 TO 21 STEP 3 ::
DISPLAY AT(A,16):S$(X);" SCH
ALE n=";TAB(27);X :: X=X+1 :
: NEXT A :: DISPLAY AT(24,1)
:" s  p  d  f   -ZUSTAND"
 :: GOTO 2700
745 !DATEN ANALYSE
746 !
747 !
780 DISPLAY AT(1,3)ERASE ALL
:"DATENSUCHE/ANALYSE
  ********************"
790 DISPLAY AT(5,4):"I) ATOM
GEWICHT?",TAB(4);"L) SDP.?",
,TAB(4);"M) SMP.?",,TAB(4);"
N) DICHTE?",TAB(4);"O) KOVAL
ENT-RADIUS?"
800 DISPLAY AT(10,4):"P) ATO
M-RADIUS?",TAB(4);"Q) IONEN-
RADIUS?",TAB(4);"R) ATOM-VOL
UMEN?",TAB(4);"S) IONISIERUN
GSENERGIE?",TAB(4);"T) EN?"
825 ON ERROR 865
830 DISPLAY AT(17,3)BEEP:"BU
CHSTB./TOLERANZ/DATENWERT" :
: ACCEPT AT(21,2)VALIDATE(UA
LPHA)SIZE(-1):AA$ :: ACCEPT
AT(21,11)VALIDATE(NUMERIC):B
 :: ACCEPT AT(21,20)VALIDATE
(NUMERIC):C :: A=ASC(AA$)-70
840 DISPLAY AT(8,4)BEEP ERAS
E ALL:"1. AUSWERTUNG HAUPTGR
```

```
UPPEN      +NEBENGRUPPEN",,TA
B(4);"2. AUSWERTUNG FUER
          LANTHANIDE U. ACTIN
IDE" :: ACCEPT AT(18,1)VALID
ATE("1,2")SIZE(-1):D
860 CALL CLEAR :: ON D GOSUB
 870,880 :: GOTO 125
865 CALL ERR(S,L):: IF S=74
THEN RETURN 780
870 Z=75 :: IF A$(1,2)="H" T
HEN 890 :: Y=2 :: GOTO 1060
880 IF A$(1,2)="CE" THEN 890
 :: Y=2 :: P=1 :: GOTO 1090
890 DISPLAY AT(24,3):"AUSDRU
CKEN ? (J/N) "
892 CALL KEY(0,K,S):: IF S=0
 THEN 892 :: IF K=74 THEN 25
00
894 CALL CLEAR :: D=1 :: FOR
 I=1 TO V
895 S=0 :: IF A$(I,A)="" THE
N 970
900 IF A=11 THEN 1010 :: W=V
AL(A$(I,A)):: IF W=C OR W>(C
-B)AND W<(C+B)THEN 950 ELSE
IF I=V THEN 955 ELSE 970
950 D=D+3 :: DISPLAY AT(D,2)
:A$(I,1):TAB(13);A$(I,A)
951 IF C=W OR T=C OR S=C THE
N CALL HCHAR(D,2,30,1)
954 IF D<>16 THEN 970 ELSE 9
55
955 DISPLAY AT(24,3):"MIT LE
ERTASTE WEITER"
960 CALL KEY(0,K,S):: IF S=0
 THEN 960 :: CALL CLEAR :: D
=1 :: IF (P=1 AND I=29)OR I=
V+1 THEN 984
970 NEXT I :: IF I=V+1 THEN
955
982 CALL KEY(0,K,S):: IF S=0
 THEN 982 :: IF K=32 THEN 98
4
984 RETURN
1010 U$=A$(I,A):: T=VAL(SEG$
(U$,1,(POS(U$,"(",1))-1))::
IF LEN(U$)<9 THEN 1030
1020 S=VAL(SEG$(U$,(POS(U$,"
(",1)+4),LEN(U$)-7-(POS(U$,"
(",1))))
1030 IF T=C OR S=C THEN 950
:: IF B=0 THEN 970 :: IF T>(
C-B)AND T<(C+B)OR S>(C-B)AND
 S<(C+B)THEN 950 ELSE 970
1035 !UNTERPROGRAMM
1036 !
1037 !
1050 IF A$(1,2)="H" THEN 210
 :: Y=1 :: GOTO 1060
1060 RESTORE 1120 :: V=75
1070 CALL CLEAR :: PRINT TAB
(2);"DATEN WERDEN GELESEN.."
 :: FOR I=1 TO V :: FOR J=1
TO 16 :: READ A$(I,J):: NEXT
 J :: NEXT I :: ON Y GOTO 21
0,890
1080 IF A$(1,2)="CE" THEN 21
0 :: Y=1
1090 PRINT "  MOMENT BITTE..
" :: FOR I=28 TO 75 :: FOR J
=1 TO 16 :: A$(I,J)="" :: NE
XT J :: NEXT I :: RESTORE 19
70 :: V=28 :: GOTO 1070
1100 FOR A=9+X TO 23-Y :: FO
R B=3 TO 30-Z STEP 3 :: CALL
 HCHAR(A,B,124):: NEXT B ::
NEXT A :: RETURN
1105 CALL SCREEN(F):: DISPLA
Y AT(1,5)ERASE ALL:"PERIODEN
SYSTEM'DER'ELEMENTE":"******
**********************" :: DI
SPLAY AT(5,9):U$ :: RETURN
1110 DATA NAME,SYMBOL,ATOMG.
,OX-ZAHL,AGGREGAT,SDP.,SMP.,
DICHTE,KOV-RADIUS,A-RADIUS,I
-RADIUS,A-VOLUMEN,I-ENERGIE,
EN,KRISTALL,S/B-CHAR.
1120 DATA WASSERSTOFF,H,1.00
797,1,G,-252.7,-259.2,0.071,
0.32,,2.08(-1),14.1,313,2.1,
HEX,S&B
1130 DATA HELIUM,HE,4.0026,0
,G,-268.9,-269.7,0.126,0.93,
,,31.8,567,,HEX,0
1140 DATA LITHIUM,LI,6.939,1
,S,1330,180.5,0.53,1.23,1.55
,0.6(+1),13.1,124,1,KUB/R,B(
+)
1160 DATA BERYLLIUM,BE,9.012
2,2,S,2770,1277,1.85,0.9,1.1
2,0.31(+2),5,215,1.5,HEX,S&B
1170 DATA BOR,B,10.811,3,S,,
2030,2.34,0.82,0.98,0.2(+3),
4.6,191,2,HEX,S
1190 DATA KOHLENSTOFF,C,12.0
1115,+-462,S,4830,3727,2.26,
0.77,0.914,2.6(-4)0.15(+4),5
.3,260,2.5,HEX,S
1200 DATA STICKSTOFF,N,14.00
67,+-3650462,G,-195.8,-210,0
.81,0.75,0.92,1.71(-3)0.11(+
5),17.3,336,3,HEX,S(+)
```

```
1220 DATA SAUERSTOFF,O,15.99
94,-2,G,-183,-218.8,1.14,0.7
3,,1.4(-2)0.09(+6),14,314,3.
5,KUB,O
1240 DATA FLUOR,F,18.9984,-1
,G,-188.2,-219.6,1.505,0.72,
,1.36(-1)0.07(+7),17.1,402,4
,,O
1250 DATA NEON,NE,20.183,,G,
-246,-248.6,1.2,0.71,,,16.8,
497,,KUB/F,O
1270 DATA NATRIUM,NA,22.9898
,1,S,892,97.8,0.97,1.54,1.9,
0.95(+1),23.7,119,0.9,KUB/R,
B(+)
1280 DATA MAGNESIUM,MG,24.31
2,2,S,1107,650,1.74,1.36,1.6
,0.65(+2),14,176,1.2,HEX,B(+
)
1300 DATA ALUMINIUM,AL,26.98
15,3,S,2450,660,2.7,1.18,1.4
3,0.5(+3),10,138,1.5,KUB/F,S
&B
1320 DATA SILICIUM,SI,28.086
,4,S,2680,1410,2.33,1.11,1.3
2,2.71(-1)0.41(+4),12.1,188,
1.8,DIAM,S&B
1330 DATA PHOSPHOR,P,30.9738
,+-3ö5ö4,S,280,44.2,1.82,1.0
6,1.28,2.12(-3)0.34(+5),17,2
54,2.1,KUB,S
1350 DATA SCHWEFEL,S,32.064,
+-2ö4ö6,S,444.6,119,2.07,1.0
2,1.27,1.84(-2)0.29(+6),15.5
,239,2.5,O/RHOMB,S(+)
1360 DATA CHLOR,CL,35.453,+-
1ö3ö5ö7,G,-34.7,-101,1.56,0.
99,,1.81(-1)0.26(+7),18.7,30
0,3,TETRAG,S(+)
1380 DATA ARGON,AR,39.948,,G
,-185.8,-189.4,1.4,0.98,,,24
.2,363,,KUB/F,O
1400 DATA KALIUM,K,39.102,1,
S,760,63.7,0.86,2.03,2.35,1.
33(+1),45.3,100,0.8,KUB/R,B(
+)
1410 DATA CALCIUM,CA,40.08,2
,S,1440,838,1.55,1.74,1.97,0
.99(+2),29.9,141,1,KUB/F,B(+
)
1420 DATA SCANDIUM,SC,44.956
,3,S,2730,1539,3,1.44,1.62,0
.81(+3),15,151,1.3,HEX,B
1430 DATA TITANIUM,TI,47.9,4
ö3,S,3260,1668,4.51,1.32,1.4
7,0.9(+2)0.68(+4),10.6,158,1
.5,HEX,S&B
1440 DATA VANADIUM,V,50.942,
5ö4ö3ö2,S,3450,1900,6.1,1.22
,1.34,0.74(+3)0.59(+5),8.35,
156,1.6,KUB/R,S&B
1450 DATA CHROM,CR,51.996,6ö
3ö2,S,2665,1875,7.19,1.18,1.
3,0.69(+3)0.52(+6),7.23,156,
1.6,KUB/R,S(+)
1460 DATA MANGAN,MN,54.938,7
ö6ö4ö2ö3,S,2150,1245,7.43,1.
17,1.35,0.8(+2)0.46(+7),7.39
,171,1.5,KUB,S(+)
1470 DATA EISEN,FE,55.847,2ö
3,S,3000,1536,7.86,1.17,1.26
,0.76(+2)0.64(+3),7.1,182,1.
8,KUB/R,S&B
1480 DATA COBALT,CO,58.933,2
ö3,S,2900,1495,8.9,1.16,1.25
,0.74(+2)0.63(+3),6.7,181,1.
8,HEX,S&B
1490 DATA NICKEL,NI,58.71,2ö
3,S,2730,1453,8.9,1.15,1.24,
0.72(+2)0.62(+3),6.6,176,1.8
,KUB/F,B
1500 DATA KUPFER,CU,63.54,2ö
1,S,2595,1083,8.96,1.17,1.28
,0.96(+1)0.69(+2),7.1,178,1.
9,KUB/F,B
1510 DATA ZINK,ZN,65.37,2,S,
906,419.5,7.14,1.25,1.38,0.7
4(+2),9.2,216,1.6,HEX,S&B
1520 DATA GALLIUM,GA,69.72,3
,L,2237,29.8,5.91,1.26,1.41,
1.13(+1)0.62(+3),11.8,138,1.
6,O/RHOMB,S&B
1530 DATA GERMANIUM,GE,72.59
,4,S,2830,937.4,5.32,1.22,1.
37,0.93(+2)0.53(+4),13.6,187
,1.8,DIAM,S&B
1540 DATA ARSEN,AS,74.922,+-
3ö5,S,613,817,5.72,1.2,1.39,
2.22(-3)0.47(+5),13.1,231,2,
RHOMB,S
1550 DATA SELEN,SE,78.96,-2ö
4ö6,S,685,217,4.79,1.16,1.4,
1.98(-2)0.42(+6),16.5,225,2.
4,HEX,S(+)
1560 DATA BROM,BR,79.909,+-1
ö5,L,58,-7.2,3.12,1.14,,1.95
(-1)0.39(+7),23.5,273,2.8,O/
RHOMB,S(+)
1570 DATA KRYPTON,KR,83.8,,G
,-152,-157.3,2.6,1.12,,,32.2
,323,,KUB/F,O
1580 DATA RUBIDIUM,RB,85.47,
1,S,688,38.9,1.53,2.16,2.48,
1.48(+1),55.9,96,0.8,KUB/R,B
(+)
```

```
1590 DATA STRONTIUM,SR,87.62
,2,S,1380,768,2.6,1.91,2.15,
1.13(+2),33.7,131,1,KUB/F,B(
+)
1600 DATA YTTRIUM,Y,88.905,3
,S,2927,1509,4.47,1.62,1.78,
0.93(+3),19.8,152,1.3,HEX,B
1610 DATA ZIRKONIUM,ZR,91.22
,4,S,3580,1852,6.49,1.45,1.6
,0.8(+4),14.1,160,1.4,HEX,S&
B
1620 DATA NIOB,NB,92.906,5ö3
,S,3300,2468,8.4,1.34,1.46,0
.7(+5),10.8,156,1.6,KUB/R,S
1630 DATA MOLYBDAEN,MO,95.94
,6ö5ö4ö3ö2,S,5560,2610,10.2,
1.3,1.39,0.68(+4)0.62(+6),9.
4,166,1.8,KUB/R,S(+)
1640 DATA TECHNETIUM,TC,98,7
,SY,,2140,11.5,1.27,1.36,,,1
67,1.9,,S(+)
1650 DATA RUTHENIUM,RU,101.0
7,2ö3ö4ö6ö8,S,4900,2500,12.2
,1.25,1.34,0.69(+3)0.67(+4),
8.3,173,2.2,HEX,S
1660 DATA RHODIUM,RH,102.905
,2ö3ö4,S,4500,1966,12.4,1.25
,1.34,0.86(+2),8.3,178,2.2,K
UB/F,S&B
1670 DATA PALLADIUM,PD,106.4
,2ö4,S,3980,1552,12,1.28,1.3
7,0.86(+2),8.9,192,2.2,KUB/F
,B
1680 DATA SILBER,AG,107.87,1
,S,2210,960.8,10.5,1.34,1.44
,1.26(+1),10.3,175,1.9,KUB/F
,S&B
1690 DATA CADMIUM,CD,112.4,2
,S,765,320.9,8.65,1.48,1.54,
0.97(+2),13.1,207,1.7,HEX,B
1700 DATA INDIUM,IN,114.82,3
,S,2000,156.2,7.31,1.44,1.66
,1.32(+1)0.81(+3),15.7,133,1
.7,TETRAG,S&B
1710 DATA ZINN,SN,118.69,4ö2
,S,2270,231.9,7.3,1.41,1.62,
1.12(+2)0.71(+4),16.3,169,1.
8,TETRAG,S&B
1720 DATA ANTIMON,SB,121.75,
-+3ö5,S,1380,630.5,6.62,1.4,
1.59,2.45(-3)0.62(+5),18.4,1
99,1.9,RHOMB,S
1730 DATA TELLUR,TE,127.6,-2
ö4ö6,S,989.8,449.5,6.24,1.36
,1.6,2.21(-2)0.56(+6),20.5,2
08,2.1,HEX,S
1740 DATA IOD,I,126.904,+-1ö
5ö7,S,183,113.7,4.94,1.33,,2
.16(-1)0.5(+7),25.7,241,2.5,
O/RHOMB,S(+)
1750 DATA XENON,XE,131.3,,G,
-108,-111.9,3.06,1.31,,,42.9
,280,,KUB/F,O
1760 DATA CAESIUM,CS,132.905
,1,L,690,28.7,1.9,2.35,2.67,
1.69(+1),70,90,0.7,KUB/R,B(+
)
1770 DATA BARIUM,BA,137.34,2
,S,1640,714,3.5,1.98,2.22,1.
35(+2),39,120,0.9,KUB/R,B(+)
1780 DATA LANTHAN,LA,138.91,
3,S,3470,920,6.17,1.69,1.87,
1.15(+3),22.5,129,1.1,HEX,B(
+)
1790 DATA HAFNIUM,HF,178.49,
4,S,5400,2222,13.1,1.44,1.67
,0.81(+4),13.6,127,1.3,HEX,S
&B
1800 DATA TANTAL,TA,180.948,
5,S,5425,2996,16.6,1.34,1.49
,0.73(+5),10.9,138,1.5,KUB/R
,S
1810 DATA WOLFRAM,W,183.85,6
ö5ö4ö3ö2,S,5930,3410,19.3,1.
3,1.41,0.64(+4)0.68(+6),9.53
,184,1.7,KUB/R,S
1820 DATA RHENIUM,RE,186.2,7
ö6ö4ö2ö-1,S,5900,3180,21,1.2
8,1.37,,8.85,182,1.9,HEX,S
1830 DATA OSMIUM,OS,190.2,2ö
3ö4ö6ö8,S,5500,3000,22.6,1.2
6,1.35,0.69(+4),8.43,201,2.2
,HEX,S
1840 DATA IRIDIUM,IR,192.2,2
ö3ö4ö6,S,5300,2454,22.5,1.27
,1.36,0.66(+4),8.54,212,2.2,
KUB/F,B
1850 DATA PLATIN,PT,195.09,2
ö4,S,4530,1769,21.4,1.3,1.39
,0.96(+2),9.1,207,2.2,KUB/F,
B
1860 DATA GOLD,AU,196.967,3ö
1,S,2970,1063,19.3,1.34,1.46
,1.37(+1),10.2,213,2.4,KUB/F
,S&B
1870 DATA QUECKSILBER,HG,200
.59,2ö1,L,357,-38.4,13.6,1.4
9,1.57,1.1(+2),14.8,241,1.9,
RHOMB,B
1880 DATA THALLIUM,TL,204.37
,3ö1,S,1457,303,11.85,1.48,1
.71,1.4(+1)0.95(+3),17.2,141
,1.8,HEX,B
```

```
1890 DATA BLEI,PB,207.19,462
,S,1725,327.4,11.4,1.47,1.75
,1.2(+2)0.84(+4),18.3,171,1.
8,KUB/F,S&B
1900 DATA BISMUT,BI,208.980,
385,S,1560,271.3,9.8,1.46,1.
7,1.2(+3)0.74(+5),21.3,185,1
.9,RHOMB,S
1910 DATA POLONIUM,PO,210,26
4,S,,254,9.2,1.46,1.76,,22.7
,,2,MONOKLIN,S&B
1920 DATA ASTAT,AT,210,+-163
6567,S,,302,,1.45,,,,,2.2,,0
1930 DATA RADON,RN,222,,G,-6
1.8,-71,,,,,,248,,KUB/F,0
1940 DATA FRANCIUM,FR,223,1,
L,,27,,,,1.76(+1),,,0.7,KUB/
R,B(+)
1950 DATA RADIUM,RA,226,2,S,
,700,5,,,1.4(+2),45,,0.9,,B(
+)
1960 DATA ACTINIUM,AC,227,3,
S,,1050,,,1.88,1.18(+3),,,1.
1,,0
1970 DATA CER,CE,140.12,364,
S,3468,795,6.67,1.65,1.81,1.
11(+2)1.01(+4),21,159,1.1,KU
B/F,B
1980 DATA PRASEODYM,PR,140.9
07,364,S,3127,935,6.77,1.65,
1.82,1.09(+3)0.92(+4),20.8,1
33,1.1,HEX,B
1990 DATA NEODYM,ND,144.24,3
,S,3027,1024,7,1.64,1.82,1.0
8(+3),20.6,145,1.2,HEX,B
2000 DATA PROMETHIUM,PM,147,
3,SY,,1027,,1.63,,1.06(+3),,
133,,HEX,B
2010 DATA SAMARIUM,SM,150.35
,362,S,1900,1072,7.54,1.62,1
.81,1.04(+3),19.9,129,1.2,RH
OMB,B
2020 DATA EUROPIUM,EU,151.96
,362,S,1439,826,5.26,1.85,1.
99,1.12(+2),28.9,131,,KUB/R,
B
2030 DATA GADOLINIUM,GD,157.
25,3,S,3000,1312,7.89,1.61,1
.79,1.02(+3),19.9,142,1.1,HE
X,B
2040 DATA TERBIUM,TB,158.924
,364,S,2800,1356,8.27,1.59,1
.80,1(+3),19.2,155,1.2,HEX,B
2050 DATA DYSPROSIUM,DY,162.
5,3,S,2600,1407,8.54,1.59,1.
80,0.99(+3),19,157,,HEX,B
2060 DATA HOLMIUM,HO,164.93,
3,S,2600,1461,8.8,1.58,1.79,
0.97(+3),18.7,,1.2,HEX,B
2070 DATA ERBIUM,ER,167.26,3
,S,2900,1497,9.05,1.57,1.78,
0.96(+3),18.4,,1.2,HEX,B
2080 DATA THULIUM,TM,168.934
,362,S,1727,1545,9.33,1.56,1
.77,0.95(+3),18.1,,1.2,HEX,B
2090 DATA YTTERBIUM,YB,173.0
4,362,S,1427,824,6.98,,1.94,
1.13(+2)0.94(+3),24.8,143,1.
1,KUB/F,B
2100 DATA LUTETIUM,LU,174.97
,3,S,3327,1652,9.84,1.56,1.7
5,0.93(+3),17.8,115,1.2,HEX,
B
2110 DATA THORIUM,TH,232.038
,4,S,3850,1750,11.7,1.65,1.8
,1.14(+3)0.95(+4),19.9,,1.3,
KUB/F,B
2120 DATA PROTACTINIUM,PA,23
1,564,S,,1230,15.4,,1.61,1.1
2(+3)0.98(+4),15,,1.5,,B
2130 DATA URAN,U,238.03,66566
463,S,3818,1132,19.07,1.42,1
.38,1.11(+3)0.97(+4),12.5,,1
.7,0/RHOMB,S&B
2140 DATA NEPTUNIUM,NP,237,6
656463,SY,,637,19.5,,1.3,1.0
9(+3)0.95(+4),21.1,,1.3,,S&B
2150 DATA PLUTONIUM,PU,242,6
656463,SY,3235,640,,,1.51,1.
07(+3)0.93(+4),,,1.3,,S&B
2160 DATA AMERICIUM,AM,243,6
656463,SY,,,11.7,,1.73,1.06(
+3)0.92(+4),20.8,,1.3,,0
2170 DATA CURIUM,CM,247,3,SY
,,,,,,,,,,,
2180 DATA BERKELIUM,BK,247,4
63,SY,,,,,,,,,,,
2190 DATA CALIFORNIUM,CF,249
,3,SY,,,,,,,,,,,
2200 DATA EINSTEINIUM,ES,254
,,SY,,,,,,,,,,,
2210 DATA FERMIUM,FM,253,,SY
,,,,,,,,,,,
2220 DATA MENDELEVIUM,MD,256
,,SY,,,,,,,,,,,
2230 DATA NOBELIUM,NO,254,,S
Y,,,,,,,,,,,
2240 DATA LAWRENCIUM,LR,257,
,SY,,,,,,,,,,,
2250 DATA K,L,M,N,O,P,Q
2260 CALL CLEAR :: END
2390 ! DRUCKEN
```

```
2391 !
2392 !
2400 RESTORE 2470 :: FOR J=1
 TO 8 :: READ DI$(J):: NEXT
J :: OPEN #4:"PIO.CR"
2410 PRINT #4: : : : :CHR$(1
4);CHR$(27);"E";TAB(12);"PER
IODENSYSTEM";CHR$(10);CHR$(1
0);CHR$(14);CHR$(27);"E";TAB
(14);"DER";CHR$(10);CHR$(10)
2420 PRINT #4:CHR$(14);CHR$(
27);"E";TAB(15);"ELEMENTE";C
HR$(10);CHR$(10);CHR$(10)
2430 PRINT #4: : : : :CHR$(2
7);"4";CHR$(15);TAB(10);B$(1
);": ";A$(I,1);CHR$(10);CHR$
(10)
2440 PRINT #4:CHR$(27);"5";C
HR$(27);"E";CHR$(27);"D";CHR
$(1);CHR$(21);CHR$(0):: FOR
J=2 TO 5
2450 PRINT #4:CHR$(10);CHR$(
9);B$(J);CHR$(9);A$(I,J):: N
EXT J :: PRINT #4:CHR$(10);C
HR$(27);"D";CHR$(1);CHR$(21)
;CHR$(42);CHR$(0):: FOR J=6
TO 13
2452 PRINT #4:CHR$(10);CHR$(
9);B$(J);CHR$(9);A$(I,J);CHR
$(9);DI$(J-5):: NEXT J :: PR
INT #4:CHR$(10);CHR$(27);"D"
;CHR$(1);CHR$(21);CHR$(0)::
FOR J=14 TO 16
2454 PRINT #4:CHR$(10);CHR$(
9);B$(J);CHR$(9);A$(I,J):: N
EXT J :: PRINT #4:CHR$(10);C
HR$(10);CHR$(10)
2460 CLOSE #4 :: FOR J=1 TO
8
2465 DI$(J)="" :: NEXT J ::
GOTO 310
2470 DATA grad celsius,grad
celsius,g/ml,angstroem,angst
roem,angstroem,w/d,kcal/g*mo
l
2490 ! DRUCKEN
2491 !
2492 !
2500 OPEN #5:"PIO.CR" :: S=0
2510 PRINT #5: : : : : : :CH
R$(14);CHR$(27);"E";TAB(25);
"DATENANALYSE";CHR$(10);CHR$
(10)
2520 PRINT #5:TAB(10);"DIE S
UCHKRITERIEN WAREN :";CHR$(1
0);CHR$(10)
2530 PRINT #5:CHR$(18);"I) A
TOMGEWICHT";CHR$(10);"L) SIE
DEPUNKT";CHR$(10);"M) SCHMEL
ZPUNKT";CHR$(10);"N) DICHTE"
2535 PRINT #5:CHR$(10);"O) K
OVALENTRADIUS";CHR$(10);"P)
ATOMRADIUS";CHR$(10)
2540 PRINT #5:"Q) IONENRADIU
S";CHR$(10);"R) ATOMVOLUMEN"
;CHR$(10);"S) IONISIERUNGSEN
ERGIE";CHR$(10);"T) ELEKTRON
EGATIVITAET"
2550 PRINT #5:CHR$(10);CHR$(
10);CHR$(27);"D";CHR$(17);CH
R$(30);CHR$(43);CHR$(0)
2552 PRINT #5:CHR$(9);"BUCHS
TABE /";CHR$(9);" TOLERANZ /
";CHR$(9);" DATENWERT";CHR$(
10);CHR$(10)
2554 PRINT #5:"IHRE EINGABE=
>";CHR$(9);AA$;CHR$(9);B;CHR
$(9);C
2560 PRINT #5:CHR$(10);CHR$(
10);CHR$(27);"4";"AUSWERTUNG
:";CHR$(10);CHR$(10);CHR$(27
);"5"
2564 PRINT #5:CHR$(27);"D";C
HR$(1);CHR$(20);CHR$(0)
2565 FOR I=1 TO V :: S=0 ::
IF A$(I,A)="" THEN 2577
2566 IF A=11 THEN 2579 :: W=
VAL(A$(I,A)):: IF W=C OR W>(
C-B)AND W<(C+B)THEN 2575 ELS
E 2577
2575 PRINT #5:CHR$(10);CHR$(
9);A$(I,1);CHR$(9);A$(I,A)
2577 NEXT I :: IF I=V+1 THEN
 2585
2579 U$=A$(I,A):: T=VAL(SEG$
(U$,1,(POS(U$,"(",1))-1)):
IF LEN(U$)<9 THEN 2581
2580 S=VAL(SEG$(U$,(POS(U$,"
(",1)+4),LEN(U$)-7-(POS(U$,"
(",1))))
2581 IF T=C OR S=C THEN 2575
 :: IF B=0 THEN 2577 :: IF T
>(C-B)AND T<(C+B)OR S>(C-B)A
ND S<(C+B)THEN 2575 ELSE 257
7
2585 PRINT #5:CHR$(27);"F";C
HR$(10);CHR$(10);CHR$(10)::
CLOSE #5 :: GOTO 984
2590 ! DRUCKEN
2591 !
2592 !
2600 OPEN #6:"PIO.CR"
```

```
2610 PRINT #6:CHR$(10);CHR$(
10);CHR$(10);CHR$(14);CHR$(2
7);"E";TAB(25);"ELEKTRONENST
RUKTUR";CHR$(20);CHR$(10);CH
R$(10);CHR$(10)
2620 PRINT #6:CHR$(27);"4";C
HR$(15);TAB(10);B$(1);": ";A
$(I,1);CHR$(10);CHR$(10);CHR
$(10);CHR$(10);CHR$(27);"5"
2630 PRINT #6:CHR$(27);"-";C
HR$(1);CHR$(27);"D";CHR$(1);
CHR$(30);CHR$(0)
2631 PRINT #6:CHR$(9);T$(1);
CHR$(9);S$(1);" SCHALE n=1";
CHR$(10);CHR$(10);CHR$(27);"
D";CHR$(1);CHR$(5);CHR$(30);
CHR$(0):: IF T$(2)="" THEN 2
647
2632 PRINT #6:CHR$(9);T$(2);
CHR$(9);T$(8);CHR$(9);S$(2);
" SCHALE n=2";CHR$(10);CHR$(
10);CHR$(27);"D";CHR$(1);CHR
$(5);CHR$(9);CHR$(30);CHR$(0
):: IF T$(3)="" THEN 2647
2633 PRINT #6:CHR$(9);T$(3);
CHR$(9);T$(9);CHR$(9);T$(14)
;CHR$(9);S$(3);" SCHALE n=3"
;CHR$(10);CHR$(10):: IF T$(4
)="" THEN 2647
2634 PRINT #6:CHR$(27);"D";C
HR$(1);CHR$(5);CHR$(9);CHR$(
14);CHR$(30);CHR$(0)
2635 PRINT #6:CHR$(9);T$(4);
CHR$(9);T$(10);CHR$(9);T$(15
);CHR$(9);T$(19);CHR$(9);S$(
4);" SCHALE n=4";CHR$(10);CH
R$(10):: IF T$(5)="" THEN 26
47
2640 PRINT #6:CHR$(9);T$(5);
CHR$(9);T$(11);CHR$(9);T$(16
);CHR$(9);T$(20);CHR$(9);S$(
5);" SCHALE n=5";CHR$(10);CH
R$(10):: IF T$(6)="" THEN 26
47
2641 PRINT #6:CHR$(27);"D";C
HR$(1);CHR$(5);CHR$(9);CHR$(
30);CHR$(0)
2642 PRINT #6:CHR$(9);T$(6);
CHR$(9);T$(12);CHR$(9);T$(17
);CHR$(9);S$(6);" SCHALE n=6
";CHR$(10);CHR$(10):: IF T$(
7)="" THEN 2647
2644 PRINT #6:CHR$(27);"D";C
HR$(1);CHR$(5);CHR$(30);CHR$
(0)
2645 PRINT #6:CHR$(9);T$(7);
CHR$(9);T$(13);CHR$(9);S$(7)
;" SCHALE n=7";CHR$(10);CHR$
(10)
2647 PRINT #6:CHR$(27);"^";C
HR$(0);"********************
";CHR$(10)
2648 PRINT #6:CHR$(27);"D";C
HR$(1);CHR$(5);CHR$(9);CHR$(
14);CHR$(30);CHR$(0)
2649 PRINT #6:CHR$(9);"s";CH
R$(9);"p";CHR$(9);"d";CHR$(9
);"f";CHR$(9);" -ZUSTAND";CH
R$(10);CHR$(10);CHR$(10);CHR
$(10);CHR$(10);CHR$(10);CHR$
(27);"F";
2650 CLOSE #6 :: DE$="" :: F
OR T=1 TO 22 :: T$(T)="" ::
NEXT T :: GOTO 125
2700 !E-TABELLE
2710 !
2720 !
2725 ON ERROR 2760
2730 OPEN #1:"DSK1."&A$(I,2)
,SEQUENTIAL,INTERNAL,VARIABL
E 22,INPUT :: FOR T=1 TO 22
:: INPUT #1:T$(T):: NEXT T
2740 X=0 :: FOR A=1 TO 7 ::
DISPLAY AT(A+1+A+X,2)SIZE(1)
:T$(A):: X=X+1 :: NEXT A ::
X=0 :: FOR A=1 TO 6 :: DISPL
AY AT(A+A+X+4,5)SIZE(1):T$(A
+7):: X=X+1 :: NEXT A
2750 X=0 :: FOR A=1 TO 5 ::
DISPLAY AT(A+A+X+7,8)SIZE(2)
:T$(A+13):: X=X+1 :: NEXT A
:: X=0 :: FOR A=1 TO 4 :: DI
SPLAY AT(A+A+X+10,12)SIZE(2)
:T$(A+18):: X=X+1 :: NEXT A
2760 CALL ERR(C,L):: IF C=13
0 AND L=1 THEN RETURN 125
2765 CLOSE #1 :: IF DE$<>"P"
 THEN 2770 ELSE 2600
2770 FOR T=1 TO 22 :: T$(T)=
"" :: NEXT T :: CALL KEY(0,K
,S):: IF S=0 THEN 2770 ELSE
125
```

```
AC   2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  1 * 14  * * *

AL   2 2 2 * * * * 6 1 * * * *
     * * * * * * * * *

AR   2 2 2 * * * * 6 6 * * * *
     * * * * * * * * *

AT   2 2 2 2 2 2 * 6 6 6 6 5 *
     10  10  10  * * 14  * * *

B    2 2 * * * * * 1 * * * * *
     * * * * * * * * *

BE   2 2 * * * * * * * * * * *
     * * * * * * * * *

BK   2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  1 * 14  8 * *

C    2 2 * * * * * 2 * * * * *
     * * * * * * * * *

CD   2 2 2 2 2 * * 6 6 6 * * *
     10  10  * * * * * * *

CF   2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  1 * 14  9 * *

CM   2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  1 * 14  7 * *

CR   2 2 2 1 * * * 6 6 * * * *
     5 * * * * * * * *

CU   2 2 2 1 * * * 6 6 * * * *
     10  * * * * * * * *

ER   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * 12  * * *

EU   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * 7 * * *

FE   2 2 2 2 * * * 6 6 * * * *
     6 * * * * * * * *

FR   2 2 2 2 2 2 1 6 6 6 6 6 *
     10  10  10  * * 14  * * *

GD   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  1 * * 7 * * *

°H   1 * * * * * * * * * * * *
     * * * * * * * * *

HF   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  2 * * 14  * * *

AG   2 2 2 2 1 * * 6 6 6 * * *
     10  10  * * * * * * *

AM   2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  * * 14  7 * *

AS   2 2 2 2 * * * 6 6 3 * * *
     10  * * * * * * * *

AU   2 2 2 2 2 1 * 6 6 6 6 * *
     10  10  10  * * 14  * * *

BA   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * * * * *

BI   2 2 2 2 2 2 * 6 6 6 6 3 *
     10  10  10  * * 14  * * *

BR   2 2 2 2 * * * 6 6 5 * * *
     10  * * * * * * * *

CA   2 2 2 2 * * * 6 6 * * * *
     * * * * * * * * *

CE   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * 2 * * *

CL   2 2 2 * * * * 6 5 * * * *
     * * * * * * * * *

CO   2 2 2 2 * * * 6 6 * * * *
     7 * * * * * * * *

CS   2 2 2 2 2 1 * 6 6 6 6 * *
     10  10  * * * * * * *

DY   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * 10  * * *

ES   2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  1 * 14  10  **

F    2 2 * * * * * 5 * * * * *
     * * * * * * * * *

FM   2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  1 * 14  11  **

GA   2 2 2 2 * * * 6 6 1 * * *
     10  * * * * * * * *

GE   2 2 2 2 * * * 6 6 2 * * *
     10  * * * * * * * *

HE   2 * * * * * * * * * * * *
     * * * * * * * * *

HG   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  10  * * 14  * * *
```

```
HO  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  * * * 11  * * *

IN  2 2 2 2 2 * * 6 6 6 1 * *
    10  10  * * * * * * *

K   2 2 2 1 * * * 6 6 * * * *
    * * * * * * * * *

LA  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  1 * * * * * *

LR  2 2 2 2 2 2 2 6 6 6 6 6 *
    10  10  10  1 * 14  14  **

MD  2 2 2 2 2 2 2 6 6 6 6 6 *
    10  10  10  1 * 14  12  **

MN  2 2 2 2 * * * 6 6 * * * *
    5 * * * * * * * *

N   2 2 * * * * * 3 * * * * *
    * * * * * * * * *

NB  2 2 2 2 1 * * 6 6 6 * * *
    10  4 * * * * * * *

NE  2 2 * * * * * 6 * * * * *
    * * * * * * * * *

NO  2 2 2 2 2 2 2 6 6 6 6 6 *
    10  10  10  1 * 14  13  **

O   2 2 * * * * * 4 * * * * *
    * * * * * * * * *

P   2 2 2 * * * * 6 3 * * * *
    * * * * * * * * *

PB  2 2 2 2 2 2 * 6 6 6 6 2 *
    10  10  10  * * 14  * * *

PM  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  * * * 5 * * *

PR  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  * * * 3 * * *

PU  2 2 2 2 2 2 2 6 6 6 6 6 *
    10  10  10  1 * 14  5 * *

RB  2 2 2 2 1 * * 6 6 6 * * *
    10  * * * * * * * *

RH  2 2 2 2 1 * * 6 6 6 * * *
    10  8 * * * * * * *

RU  2 2 2 2 1 * * 6 6 6 * * *
    10  7 * * * * * * *

I   2 2 2 2 2 * * 6 6 6 5 * *
    10  10  * * * * * * *

IR  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  7 * * 14  * * *

KR  2 2 2 2 * * * 6 6 6 * * *
    10  * * * * * * * *

LI  2 1 * * * * * * * * * * *
    * * * * * * * * * *

LU  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  1 * * 14  * * *

MG  2 2 2 * * * * 6 * * * * *
    * * * * * * * * * *

MO  2 2 2 2 1 * * 6 6 6 * * *
    10  5 * * * * * * *

NA  2 2 1 * * * * 6 * * * * *
    * * * * * * * * * *

ND  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  * * * 4 * * *

NI  2 2 2 2 * * * 6 6 * * * *
    8 * * * * * * * *

NP  2 2 2 2 2 2 2 6 6 6 6 6 *
    10  10  10  1 * 14  4 * *

OS  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  6 * * 14  * * *

PA  2 2 2 2 2 2 2 6 6 6 6 6 *
    10  10  10  1 * 14  2 * *

PD  2 2 2 2 * * * 6 6 6 * * *
    10  10  * * * * * * *

PO  2 2 2 2 2 2 * 6 6 6 6 4 *
    10  10  10  * * 14  * * *

PT  2 2 2 2 2 1 * 6 6 6 6 * *
    10  10  9 * * 14  * * *

RA  2 2 2 2 2 2 2 6 6 6 6 6 *
    10  10  10  * * 14  * * *

RE  2 2 2 2 2 2 * 6 6 6 6 * *
    10  10  5 * * 14  * * *

RN  2 2 2 2 2 2 * 6 6 6 6 6 *
    10  10  10  * * 14  * * *

S   2 2 2 * * * * 6 4 * * * *
    * * * * * * * * * *
```

```
SB   2 2 2 2 2 * * 6 6 6 3 * *
     10  10  * * * * * * *

SC   2 2 2 2 * * * 6 6 * * * *
     1 * * * * * * * *

SE   2 2 2 2 * * * 6 6 4 * * *
     10  * * * * * * * *

SI   2 2 2 * * * * 6 2 * * * *
     * * * * * * * * *

SM   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * 6 * * *

SN   2 2 2 2 2 * * 6 6 6 2 * *
     10  10  * * * * * * *

SR   2 2 2 2 2 * * 6 6 6 * * *
     10  * * * * * * * *

TA   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  3 * * 14  * * *

TB   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * 9 * * *

TC   2 2 2 2 2 * * 6 6 6 * * *
     10  5 * * * * * * *

TE   2 2 2 2 2 * * 6 6 6 4 * *
     10  10  * * * * * * *

TH   2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  2 * 14  * * *

TI   2 2 2 2 * * * 6 6 * * * *
     2 * * * * * * * *

TL   2 2 2 2 2 2 * 6 6 6 6 1 *
     10  10  10  * * 14  * * *

TM   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * 13  * * *

U    2 2 2 2 2 2 2 6 6 6 6 6 *
     10  10  10  1 * 14  3 * *

V    2 2 2 2 * * * 6 6 * * * *
     3 * * * * * * * *

W    2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  4 * * 14  * * *

XE   2 2 2 2 2 * * 6 6 6 6 * *
     10  10  * * * * * * *

Y    2 2 2 2 2 * * 6 6 6 * * *
     10  1 * * * * * * *

YB   2 2 2 2 2 2 * 6 6 6 6 * *
     10  10  * * * 14  * * *

ZN   2 2 2 2 * * * 6 6 * * * *
     10  * * * * * * * *

ZR   2 2 2 2 2 * * 6 6 6 * * *
     10  2 * * * * * * *
```

PERIODENSYSTEM

DER

ELEMENTE

SYMBOL	HG	NAME: QUECKSILBER
ATOMG.	200.59	
OX-ZAHL	2ö1	
AGGREGAT	L	
SDP.	357	grad celsius
SMP.	-38.4	grad celsius
DICHTE	13.6	g/ml
KOV-RADIUS	1.49	angstroem
A-RADIUS	1.57	angstroem
I-RADIUS	1.1(+2)	angstroem
A-VOLUMEN	14.8	w/d
I-ENERGIE	241	kcal/g*mol
EN	1.9	
KRISTALL	RHOMB	
S/B-CHAR.	B	

BROKER

BROKER ist ein Abenteuer, in dem es nicht um Elfen und Feen oder irgendwelche Fabelwesen geht, sondern um die harte Realität der Börse. Als Aktionär gegen zwei oder drei Mitspieler versuchen Sie, Ihr Startkapital so rasch wie möglich zu vermehren. Durch Börsenmanipulationen, Mehrheitskauf oder geschickte Aktionen tricksen Sie Ihre Mitspieler aus. Mit BROKER haben Sie ein aufwendiges, außergewöhnlich anspruchsvolles Spiel, welches besondere Anforderungen an Ihr kalkulatorisches Geschick stellt.

ABTIPPEN: Wegen seiner Länge ist BROKER in drei Blocks aufgeteilt. Tippen Sie zunächst den Lader ab und speichern Sie ihn unter einem beliebigen Programmnamen – am Besten unter "LOAD". Getestet werden kann das Programm erst, wenn alle drei Teile auf einer Diskette vorliegen.

Teippen Sie danach Teil I ab und speichern Sie ihn unter dem Namen "BROKER-I". Sodann verfahren Sie genauso mit Teil II unter dem Namen "BROKER-II".

Wenn Sie den beiden Hauptteilen andere Namen verwenden, können sich die Teile untereinander nicht gegenseitig aufrufen.

In den Hauptteilen werden einige Sonderzeichen verwendet, die Sie durch (CTRL) + (TASTE) darstellen können:

Buchstabe è: CHR$ (129) (CTRL) + (A)

Verkehrtes ?: CHR$ (136) (CTRL) + (H)

Ansonsten tippen Sie bitte das Programm wie gedruckt ab. Der Hauptteil benutzt die gesamte Programmspeicherkapazität und die Veränderung oder das Hinzufügen von Zeilen kann den Ablauf beeinflussen.

Wer das Programm nicht abtippen möchte, kann es als Leser dieser Zeitschrift günstig direkt bei Rausch & Haub bestellen. Adresse siehe Listing!

Spielablauf: Teil I des Programms beinhaltet eine Bedienungsanleitung in Kurzfassung. Das komplette Handbuch einschließlich Sammelordner zu diesem Programm ist gegen DM 9,50 bei der im Kopf des Listings genannten Adresse erhältlich. Die wesentlichen Funktionen:

© *by H.-G. Rausch*

BROKER – DAS SPIEL

STARTEN: Wenn der Lader mit "LOAD" abgespeichert ist, startet das Programm von selbst. Ein Titelbild erscheint und eine Melodie ertönt. Sie haben verschiedene Auswahlmöglichkeiten, ob Sie direkt mit dem Spiel beginnen wollen oder erst die Spielregeln lesen möchten. Sie werden vom Computer angezeigt.

Das Hauptprogramm (Teil II) wird vom ersten Teil mit der Spielregel geladen. Sie haben jetzt die Möglichkeit, ein neues Spiel zu beginnen oder ein altes von Diskette zu laden. Alte Spiele werden durch DATEINAME geladen; Fehler können zum Sperren des Computers führen.

Wenn Sie ein neues Spiel beginnen, müssen Sie die folgenden Anfangswerte festlegen: Spielerzahl (2 oder 3), Spielernamen (je bis 5 Großbuchstaben), Startkapital (500–12000) und Spielende bei DM ( )=10faches Startkapital). Die Standardwerte können mit (ENTER) übernommen werden. Wenn Sie einen Fehler machen, können Sie die Eingaben nach Korrektur? J–(ENTER) wiederholen. Danach ist eine Änderung der Basiswerte nicht mehr möglich.

Jetzt beginnt das Spiel: Der Computer meldet sich mit dem 'Branchenindex'. Drücken Sie (SPACE), und Sie erhalten ein Menü mit allen möglichen Funktionen:

DIE FUNKTIONEN VON BROKER:

KAPITALLISTE (1): Anzeige und Aufschlüsselung des Bruttokapitals eines Spielers (des Spielers, der an der Reihe ist).

AKTIEN KAUFEN (2): Ankauf von Aktien durch Eingabe einer AG-Nummer und einer Aktienmenge. Der Computer gibt entsprechende Anweisungen und Kommentare.

AKTIEN VERKAUFEN (3): Verkauf von Aktien durch Eingabe einer AG-Nummer und einer Aktienmenge. Der Computer gibt auch hier entsprechende Hinweise.

BÖRSENAKTIONEN (4): Zufällige Wahl einer Kursveränderung. Es gibt normale und besondere Aktionen. Aktionen können sein: Dividende und Aufwertung einzelner AG; Hausse/Baisse von einer oder vilen AG; Multiple Action Flood mit gewaltigen Kursveränderungen nach unten und oben; Manuel Activity für direkte Manipulationen an 2 Gesellschaften. Einige davon können unterbunden werden, wenn die Nachricht 'AKZEPTIEREN SIE?' erscheint. Hin und wieder dürfen Sie auch eine dieser Aktionen direkt auswählen. Statt einer Börsenaktion kann auch das FINANZAMT ausgewählt werden, welches dann eventuell Steuern vom Bargeld kassiert.

MANAGEMENT (5): Mit 25 % Aktien einer AG können Sie Verfügungen in einer AG treffen, was bedeutet, Sie entscheiden, wie Überschüsse verwendet werden, die normalerweise als Dividende an die Aktionäre ausgeschüttet werden. Auch besteht die Möglichkeit, sich genauer über eine AG zu informieren. Der Computer gibt entsprechende Hinweise.

SPIELERWECHSEL (6): Sie geben die Runde an den nächsten Spieler weiter, wenn Sie keine Aktion mehr durchführen wollen oder können. Der Computer führt einige Berechnungen aus (Dividende?, Aufwertung?, Wahl des Aufsichtsrates?
Vergütungen?, Spielende? etc.) und gibt Ihnen die Möglichkeit, das laufende Spiel abzuspeichern (mit DATEINAME).

KAPITALVERGLEICH (7): Das Kapital aller Spieler wird aufgelistet. Wenn das Nettokapital unter DM 100,– sinkt, ist das Spiel verloren!

AKTIENVERGLEICH (B): Es wird angezeigt, welcher Spieler von welcher Gesellschaft wieviele Aktien besitzt.

GRAFIKAUSWERTUNG (9): Die letzten 23 Kursbewegungen einer oder zwei AG nach Wahl werden grafisch auf dem Bildschirm dargestellt. Meßzeitpunkt für die Grafik ist jeder Spielerwechsel.

ABSPEICHERN UND EINLESEN:
- Datendisketten / Leerdiskette einlegen
- ENTER drücken
- Diskettennamen eingeben
- ENTER drücken
- Datenbezeichnung ohne Laufwerk eingeben
- ENTER drücken

ALLGEMEINE SPIELREGLEN:
Die Spieler kommen nacheinander an die Reihe. Wer an der Reihe ist, wird angezeigt. Für jede Runde stehen einem

Spieler Handlungspunkte (HP) zur Verfügung, mit denen er unterschiedliche Aktionen auswählen kann. Wieviele HP und wieviel Geld eine Aktion kostet, ist im Menü ersichtlich.
Es dürfen keine Aktien gekauft werden, die in der gleichen Runde verkauft wurden – und umgekehrt. Der Computer ignoriert derartige Versuche.
Wenn der Kurs unter DM 50,– sinkt, findet beim nächsten Spielerwechsel eine Aufwertung statt, die von den Aktionären finanziert wird. Wer nicht zahlen kann, muß seine Aktien verkaufen.
Wenn der Kurs über DM 2000,– steigt, findet beim nächsten Spielerwechsel eine Aufwertung statt. Jeder Aktioner erhält eine entsprechende Dividende.
Das Spielkapital muß durch den Kauf von Aktien und das Erhöhen des Kurses vermehrt werden. Wer als erster das Spielziel erreicht, der hat gewonnen. Das Spiel ist auch beendet, wenn ein Spieler weniger als DM 100,– besitzt. In diesem Fall gewinnt der Spieler mit dem größten Nettokapital (Briefgeld und Bargeld abzüglich Schulden).
Wenn ein Spieler alle HP verbraucht hat, kann er keine Aktionen mehr durchführen, die sein Kapital vermehren können. Er sollte dann die Runde an den nächsten Spieler weitergeben.
Es ist erlaubt, sich Notizen über den Spielverlauf zu machen.

PROGRAMMAUFBAU:
Das Listing ist mit REM-Zeilen unterteilt und gut strukturiert. Eine Variablen-Liste und eine Strukturbeschreibung können gegen DM 5,– UKB von jedem Interessenten bei den Autoren angefordert werden, da sie den Rahmen dieser Beschreibung sprengen würde.

Hans-Georg Rausch

```
100 ON BREAK NEXT
101 !XXXXXXXXXXXXXXXXXXXXXXX
102 !X                      X
103 !X      LOADER          X
104 !X                      X
105 !XXXXXXXXXXXXXXXXXXXXXXX
106 !
107 CALL CLEAR
108 ON BREAK NEXT :: ON ERROR
110
109 RUN "DSK1.BROKER-1"
110 PRINT "** LOAD ERROR **"
111 FOR I=1 TO 50 :: CALL
SOUND(10,110,100) :: FOR J=1
TO 50 :: NEXT J :: NEXT I
112 PRINT "* PRESS ENTER TO
CONTINUE! *"
113 CALL KEY(0,K,S) :: IF S=0
THEN 113
114 GOTO 110
```

```
100 ON BREAK NEXT :: ON WARNING NEXT ::
ON ERROR 100
101 ! **********************
102 ! **********************
103 ! **                    **
104 ! ** B  R  O  K  E  R **
105 ! **                    **
106 ! **********************
107 ! **********************
108 !
109 ! **********************
110 ! *                    *
111 ! *WIRTSCHAFTSABENTEUER*
112 ! * FUER DEN  TI-99/4A *
113 ! * HOMECOMPUTER SOWIE *
114 ! *  EXT.-BASIC-MODUL  *
115 ! *  DISKLAUFWERK UND  *
116 ! *  MEMORY-EXPANSION  *
117 ! *  (32Kb-CARTRIDGE)  *
118 ! **********************
119 !
120 ! **********************
121 ! *                    *
122 ! * (C)1984 BY HAGERA  *
123 ! *                    *
124 ! * HANS-GEORG RAUSCH  *
125 ! * C/O RAUSCH & HAUB  *
126 ! * POSTFACH 32 03 13  *
127 ! * 5300  BONN  3      *
128 ! *                    *
129 ! **********************
130 !
131 ! **********************
132 ! *                    *
133 ! * T  E  I  L    I *
134 ! *                    *
135 ! **********************
136 !
137 ! **********************
138 !
139 !
140 !
141 ! ** DUNKEL **
142 !
143 CALL CLEAR :: CALL SCREEN(2)
144 FOR I=1 TO 14 :: CALL COLOR(I,2,2)::
 NEXT I
145 !
146 ! ** CHARDEF **
147 !
148 CALL CHAR(129,"FFFFFFFFFFFFFFFF00000
00000000000")
149 !
150 ! ** DISPLAY **
151 !
152 RESTORE 356 :: FOR I=1 TO 24 :: READ
 A$ :: DISPLAY AT(I,1):A$ :: NEXT I
153 !
154 ! ** LESBAR  **
155 !
156 FOR I=1 TO 12 :: CALL COLOR(I,7,2)::
 NEXT I
157 CALL COLOR(13,13,5)
158 !
159 ! ** TRICK **
160 !
161 CALL COLOR(13,5,13)
162 READ D,F1,F2,F3 :: IF D<>0 THEN CALL
 SOUND(D,F1,0,F2,0,F3,0)ELSE RESTORE 385
163 CALL KEY(0,K,S):: IF S=1 THEN 168
164 CALL COLOR(13,13,5)
165 READ D,F1,F2,F3 :: IF D<>0 THEN CALL
 SOUND(D,F1,0,F2,0,F3,0)ELSE RESTORE 385
166 CALL KEY(0,K,S):: IF S=1 THEN 168
167 GOTO 161
168 ! ** TAFEL **
169 !
170 DISPLAY AT(8,3):"BITTE WAEHLEN SIE:"
171 DISPLAY AT(9,3):"<1> SPIELSTART";"
<2> SPIELREGEL + START":"  <3> SPIELENDE
":""
172 !
173 CALL KEY(0,K,S):: IF K<49 OR K>51 TH
EN 173
```

```
174 ON K-48 GOTO 175,177,182
175 DISPLAY AT(8,1):"     WIRD JETZT GEL
ADEN     ":"":"   BITTE HABEN SIE GEDULD
   ":"":"" :: DISPLAY AT(22,5)SIZE(20):R
PT$(" <C> 1984 ",2)
176 RUN "DSK1.BROKER-2"
177 CALL CLEAR :: GOSUB 183
178 CALL CLEAR :: RESTORE 356
179 FOR I=1 TO 24 :: READ A$ :: DISPLAY
AT(I,1):A$ :: NEXT I
180 DISPLAY AT(22,5)SIZE(20):"BROKER WIR
D GELADEN!"
181 RUN "DSK1.BROKER-2"
182 CALL CLEAR :: RUN 101
183 ! ** EINFUEHRUNG **
184 !
185 P=1
186 DISPLAY AT(1,21):USING "SEITE ##":P
187 ON P GOSUB 194,206,216,226,236,246,2
56,266,276,286,296,306,316,326,336,346
188 CALL KEY(0,K,S):: IF S=0 THEN 188
189 IF K=32 THEN P=P+1 :: IF P=17 THEN P
=1
190 IF K=13 THEN P=P-1 :: IF P=0 THEN P=
1
191 IF K=14 THEN RETURN
192 GOTO 186
193 !
194 ! ****SEITE 1****
195 !
196 DISPLAY AT(1,1)SIZE(6):"BROKER" :: D
ISPLAY AT(2,1):RPT$("=",28):: DISPLAY AT
(22,1):RPT$("=",28):"<ENTER>!!<SPACE>!!<
BEGIN>"
197 DISPLAY AT(24,1):"ZURUECK!!WEITER !!
SPIELSTART"
198 DISPLAY AT(4,1):""
199 DISPLAY AT(5,1):"      DAS BOERSENSP
IEL":"":"  èè  èè   è  èèè  è":" è    è
è  è  è    è"
200 DISPLAY AT(9,1):"  è   èè   è  èè
è    èèè    è  è    è  è    è":" èè   è
   è  èèè  èèè":""
201 DISPLAY AT(13,1):"  èè   èèè   èèè
èèè  è":"  è  è  è    è     è    è":"  èè
   èè   è èè  èè   è":"  è  è  è    è  è
 è    è"
202 DISPLAY AT(17,1):"  è è  èèè   èè
èèè  èèè":"":"     <C> 1984 BY HAGERA":"
"
203 !
204 RETURN
205 !
206 ! ****SEITE 2****
207 !
208 DISPLAY AT(4,1):"SPIELSINN"
209 DISPLAY AT(5,1):"Ziel von Broker ist
 es, seinStartkapital durch geschick=tes
 An- und Verkaufen diver=ser Aktien zu v
ermehren. Die"
210 DISPLAY AT(9,1):"Aktienkurse werden
durch denComputer festgelegt; einige=mal
e kann der Spieler selbstentscheiden. Du
rch  Sperrmi="
211 DISPLAY AT(13,1):"noritaeten  kann d
er Spielerauch  direkt ueber bestimmteDi
nge entscheiden. Eine Rei=he  Informatio
nen helfen dem"
212 DISPLAY AT(17,1):"Spieler bei seinen
 Entschei=dungen.":"Kalkulationsgabe und
 Risiko=bereitschaft sind gefragt."
213 !
214 RETURN
215 !
216 ! ****SEITE 3****
217 !
218 DISPLAY AT(4,1):"SPIELSTART"
219 DISPLAY AT(5,1):"Zu Beginn muessen e
inige Da=ten festgelegt werden, damitder
 Rechner weiss, wann  einSpiel beendet i
st und welche"
220 DISPLAY AT(9,1):"Geldsumme Sie besit
zen. Aus=serdem wird nach Spielerzahlund
 -namen gefragt. Da allesim Dialogverfah
ren abgefragt"
221 DISPLAY AT(13,1):"wird, ist die Eing
abe super=leicht.":"Statt  der manuellen
 Eingabeder Werte koennen Sie jedoch"
222 DISPLAY AT(17,1):"auch einen alten S
pielstand,von Diskette, einlesen. Auchhi
erbei werden alle Handlun=gen einzeln ab
verlangt."
223 !
224 RETURN
225 !
226 ! ****SEITE 4****
227 !
228 DISPLAY AT(4,1):"SPIELAUFBAU"
229 DISPLAY AT(5,1):"Ein Spiel besteht a
us sovie=len Runden, wie noetig sind,um
die Spielendsumme zu ver=dienen. In eine
r Runde kommt"
230 DISPLAY AT(9,1):"jeder Spieler einma
l an  dieReihe und kann bestimmte Ak=tio
nen durchfuehren.":"Die Spielaktionen ha
ben ver="
231 DISPLAY AT(13,1):"schiedene Wertigke
iten. Jedekostet  1-3 Handlungspunkte,vo
n denen der Spieler zu Be=ginn jeder Run
de  5 besitzt."
232 DISPLAY AT(17,1):"Manche Aktionen ko
sten  aberzusaetzlich Geld!":"Im Folgend
en  erfahren  Sie,was jede Aktion bewirk
t."
233 !
234 RETURN
235 !
236 ! ****SEITE 5****
237 !
238 DISPLAY AT(4,1):"MENUE"
239 DISPLAY AT(5,1):"Durch Druecken eine
r  belie=bigen Taste erhalten Sie dieHau
ptwahlliste (ausser 0-9).Diese zeigt, mi
t welcher Ta="
240 DISPLAY AT(9,1):"ste Sie welche Akti
on errei=chen und wieviel diese  Wertist
 (DM und Handlungspunkte)und erwartet Ih
re  Tastatur="
241 DISPLAY AT(13,1):"eingabe.":"Das Men
ue ist kostenfrei!":"Durch Druecken eine
r Ziffer=taste (0-9) koennen die ver="
242 DISPLAY AT(17,1):"schiedenen Aktione
n auch di=rekt angewaehlt werden.":"Manc
he  Eingaben muessen mit<ENTER> bestaeti
gt werden."
243 !
244 RETURN
245 !
246 ! ****SEITE 6****
247 !
248 DISPLAY AT(4,1):"BRANCHENINDEX  TAST
E <0>"
249 DISPLAY AT(5,1):"In Ihm werden die S
ortenkur=se angegeben. Er kostet kei=nen
 Handlungspunkt (HP), je=doch DM 5.-!"
250 DISPLAY AT(9,1):"":"KAPITALLISTE   T
ASTE <1>":"Hier wird fuer jeden Spielerd
as Bruttokapital ausgegeben"
251 DISPLAY AT(13,1):"(Kapital ohne Schu
ldenabzug)und zwar aufgeschluesselt inve
rbrieftes Kapital und Bar=geld."
252 DISPLAY AT(17,1):"Die Kapitalliste k
ostet kei=ne HP, jedoch ebenfalls 5.-!":
"":"Farbe: Schwarz auf gruen."
253 !
254 RETURN
255 !
256 ! ****SEITE 7****
257 !
258 DISPLAY AT(4,1):"AKTIEN KAUFEN  TAST
E <2>"
259 DISPLAY AT(5,1):"Durch Eingabe einer
 Sorte u.einer Menge  koennen  Aktienzum
 Kurswert erworben werden(vorausgesetzt,
  Ihr Kapital"
260 DISPLAY AT(9,1):"reicht aus).":"Der
```

```
Ankauf von Aktien kostet2 Handlungspunkt
e. Ausserdemwerden Maklergebuehren  - 2%
"
261 DISPLAY AT(13,1):"vom Umsatz -  bere
chnet. Siekoennen keine Sorte  kaufen,di
e Sie in der selben  Rundeerst verkauft
haben. Fehler="
262 DISPLAY AT(17,1):"hafte Eingaben wer
den  ange=zeigt und muessen wiederholtwe
rden.":"Farbe: Gruen auf grau."
263 !
264 RETURN
265 !
266 ! ****SEITE 8****
267 !
268 DISPLAY AT(4,1):"AKTIEN VERKAUFEN  T
ASTE <3>"
269 DISPLAY AT(5,1):"Durch Eingabe einer
 Sorte u.einer Menge  koennen  Aktienzum
 Kurswert verkauft werden(vorausgesetzt,
 Sie besitzen"
270 DISPLAY AT(9,1):"auch Aktien dieser
Sorte).  Die Kosten entsprechen denenbei
m Ankauf.":"Ebenso ist es nicht moeglich
"
271 DISPLAY AT(13,1):"Aktien zu verkaufe
n, die  inder selben Runde erst erwor=be
n worden sind.":"Hilfestellungen beim Ve
rkauf"
272 DISPLAY AT(17,1):"bieten die Angaben
 ueber denletzten Ankauf.":"":"Farbe: Ro
t auf grau."
273 !
274 RETURN
275 !
276 ! ****SEITE 9****
277 !
278 DISPLAY AT(4,1):"BOERSENAKTION  TAST
E <4>"
279 DISPLAY AT(5,1):"Durch Druecken der
Taste <4>erreichen Sie die zufaelligeAus
wahl einer der folgenden,in der schriftl
ichen Erlaeu=="
280 DISPLAY AT(9,1):"terung ausfuehrlich
 erklaer=ten, Boersenaktionen:":"> Divid
ende / Aufwertung (3)> Hausse - Baisse
     (1)"
281 DISPLAY AT(13,1):"> Multiple Action
Flood  (3)> Brokers Manual Activity(3) "
:"Um <4> waehlen  zu  koennen,"
282 DISPLAY AT(17,1):"sind mindestens 3
HP noetig.Abgezogen  wird  allerdingsnu
r der Wert in Klammern.":"Farbe: Je nach
 Aktionsart."
283 !
284 RETURN
285 !
286 ! ****SEITE 10****
287 !
288 DISPLAY AT(4,1):"MANAGEMENT  TASTE <
5>"
289 DISPLAY AT(5,1):"Hier erhalten Sie d
ie  Moeg=lichkeit,  Aufsichtsratsvor=sit
zender zu werden und  da=mit die Geschic
ke  einer be="
290 DISPLAY AT(9,1):"stimmten AG zu beei
nflussen.Die  vielfaeltigen Moeglich=kei
ten werden in der  beige=fuegten schrift
lichen Spiel="
291 DISPLAY AT(13,1):"regel ausfuehrlich
 behandeltund beschrieben.":"Management
kostet pro Aktion1 Handlungspunkt und DM
 10.-"
292 DISPLAY AT(17,1):"Gebuehr.  Fuer die
 Benutzungder Aktion sollte Sperrmino=ri
taet einer AG vorliegen!":"Farbe: Gelb a
uf rot."
293 !
294 RETURN
295 !
296 ! ****SEITE 11****
297 !
298 DISPLAY AT(4,1):"SPIELABGABE TASTE <6>"
299 DISPLAY AT(5,1):"Hat ein Spieler sei
ne Aktio=nen beendet,  so gibt er dasSpi
el weiter. Diese  Eingabebedarf der Best
aetigung:"
300 DISPLAY AT(9,1):"":"<N> Nein (nicht
bestaetigen)":"":"<@> Spielunterbrechung
"
301 DISPLAY AT(13,1):"     (Spielstand ab
speichern) ":"< > Bestaetigung (naechste
r      Spieler ist a.d. Reihe)"
302 DISPLAY AT(17,1):"":"Nach <@> oder <
 > wird  eineEndkontrolle durchgefuehrt.
":"Farbe: Weiss auf blau."
303 !
304 RETURN
305 !
306 ! ****SEITE 12****
307 !
308 DISPLAY AT(4,1):"KAPITALVERGLEICH  T
ASTE <7>"
309 DISPLAY AT(5,1):"Hier kann das Kapit
al  allerSpieler direkt verglichen u.kon
trolliert werden (DM 5!):":""
310 DISPLAY AT(9,1):"  Name    Bar Dm:  [
Summe der+ Verbrieft  Dm:  [Gebuehren= B
rutto      Dm:  [und ande=- Schulden   DM
:<< rer  Ver="
311 DISPLAY AT(13,1):"= Netto       DM:
[bindlich=                    [keiten.":""
:""
312 DISPLAY AT(17,1):"Das Spiel ist been
det, fallsdas Nettokapital unter 100.-fa
ellt oder der Spielendebe=trag erreicht
wird."
313 !
314 RETURN
315 !
316 ! ****SEITE 13****
317 !
318 DISPLAY AT(4,1):"AKTIENVERGLEICH  TA
STE <8>"
319 DISPLAY AT(5,1):"Hier koennen Sie se
hen, wie=viel Aktien einer Sorte  dieein
zelnen Spieler besitzen. Preis DM 5.- /
keine HP."
320 DISPLAY AT(9,1):"Farbe: Schwarz auf
gruen.":"":"GRAFIKAUSWERTUNG  TASTE <9>
Wenn Sie wissen wollen,  wie"
321 DISPLAY AT(13,1):"sich  der  Kurs  e
iner Aktieauf lange Sicht bewegt  hat,so
 koennen Sie fuer  DM 10.-jeweils eine o
der zwei AG in"
322 DISPLAY AT(17,1):"graphischer Form v
ergleichenbzw. ausgeben. Rechneranzei=ge
n beachten!":"Farbe: Weiss auf schwarz."
323 !
324 RETURN
325 !
326 ! ****SEITE 14****
327 !
328 DISPLAY AT(4,1):"SPIELENDE"
329 DISPLAY AT(5,1):"Das Spiel ist beend
et,  wennes einem Spieler gelingt,denAnf
angs festgelegten 'Spiel=endebetrag' zu
erreichen."
330 DISPLAY AT(9,1):"Verglichen wird mit
 dem Net=tokapital. Um aber gewinnen zu
koennen, darf der Spielerkeine Schulden
haben!"
331 DISPLAY AT(13,1):"Die kontrolle wird
 immer nurfuer  den  Spieler  durchge=fu
ehrt, der am Spiel ist.":""
332 DISPLAY AT(17,1):"":"Viel Erfolg bei
m brokern.":"":"":""
333 !
334 RETURN
335 !
336 ! ****SEITE 15****
337 !
338 DISPLAY AT(4,1):"RATSCHLAEGE"
339 DISPLAY AT(5,1):"Wenn Sie Broker  zu
m  erstenMal spielen, sollten Sie aufjed
en Fall die ausfuehrlicheDokumentation z
```

```
u diesem Pro="
340 DISPLAY AT(9,1):"gramm lesen, da die
se Anlei=tung nur einen kurzen Ueber=bli
ck geben  soll  und nichtalle vielfaelti
gen Moeglich="
341 DISPLAY AT(13,1):"keiten des Spiels
behandelt.Der geuebte Computerfan, derau
ch gern  einmal  den  Kopfzum Spielen be
nutzt, wird an"
342 DISPLAY AT(17,1):"langen Abenden sic
her oeftereinmal zum 'brokern'  einla=de
n und viel  Freude an die=sem Programm h
aben."
343 !
344 RETURN
345 !
346 ! ****SEITE 16****
347 !
348 DISPLAY AT(4,1):""
349 DISPLAY AT(5,1):"Fuer Anregungen und
 Tips binich immer Dankbar.":"":"Meine A
dresse:"
350 DISPLAY AT(9,1):"":"HAGERA":"Hans-Ge
org Rausch":"C/O RAUSCH & HAUB"
351 DISPLAY AT(13,1):"POSTFACH  32 03 13
":"5300 Bonn 3":"":""
352 DISPLAY AT(16,1):"Dieses und andere
Programme sind dort erhaeltlich.  Ein Ge
samtkatalog mit 52 Seiten kann gegen DM
3.50 bestellt werden."
353 !
354 RETURN
355 !
356 ! ** DATAZEILEN **
357 !
358 DATA " "
359 DATA "  èè  èè   èè  è è èèè èè   "
360 DATA "  è è è è è  è è è è   è è  "
361 DATA "  èè  èè  è  è èè  èè  èè   "
362 DATA "  è è è è è  è è è è   è è  "
363 DATA "  èè  è è  èè  è è èèè è è  "
364 DATA " "
365 DATA "  EIN SPIEL FUER DEN TI 99  "
366 DATA "  HOMECOMPUTER MIT EXBASIC  "
367 DATA "  SPEICHERERWEITERUNG  UND  "
368 DATA "  DISKETTENSTATION <C>1984  "
369 DATA "            BY"
370 DATA " "
371 DATA " à à  àà   ààà ààà àà   àà  "
372 DATA " à à à  à à    à   à à à  à "
373 DATA " ààà àààà à àà àà  àà  àààà "
374 DATA " à à à  à à  à à   à à à  à "
375 DATA " à à à  à  àà  ààà à à à  à "
376 DATA " "
377 DATA " èèèèèèèèèèèèèèèèèèèèèèèè "
378 DATA " à                      è "
379 DATA " è  BITTE TASTE DRUECKEN  à "
380 DATA " à                      è "
381 DATA " èèèèèèèèèèèèèèèèèèèèèèèè "
382 !
383 ! ** MUSIKDATA **
384 !
385 DATA -500,110,220,330
386 DATA -800,400,215,215
387 DATA -300,350,350,350
388 DATA -400,220,330,440
389 DATA -450,330,320,500
390 DATA -600,400,400,200
391 DATA -750,300,200,450
392 DATA -500,250,300,450
393 DATA -320,400,600,800
394 DATA -600,720,520,320
395 DATA -800,300,600,900
396 DATA -250,110,220,330
397 DATA 0,0,0,0
```

```
100 ! ON BREAK NEXT
101 ! ******************
102 ! *                *
103 ! * BROKER TEIL II  *
104 ! *                *
105 ! ******************
106 ! INITIALISIERUNG
107 CALL CLEAR :: OPTION BASE 1
108 DIM A$(25),Z(11),V(11),X(11),W(3,11)
,NOTE$(11),ST$(11),D(3),K(3),P$(4),S(3),
LK$(3),R(11)
109 P$(4)="BANK"
110 RANDOMIZE :: ON WARNING NEXT
111 IMAGE ########## ##### #### #####
112 IMAGE ##. ########## ### # #####
113 IMAGE #### ####
114 CALL CHAR(91,"006EAA2A2A2A2E00",93,"
00060A0202020200",129,"003838387C3810",1
36,"0010387C383838")
115 CALL CHAR(130,"0000183C3C180000",137
,"0000183C3C180000")
116 CALL COLOR(12,16,2,13,7,2,14,13,2)::
 CALL CHCOL(7,15):: GOTO 366
117 DISPLAY AT(1,3)ERASE ALL BEEP:"B  R
 O  K  E  R  -  III": :RPT$("*",28): :"S
PIELERZAHL...: 2":"(OPTION: 3)"
118 DISPLAY AT(8,1):"NAME SPIELER 1: "&P
$(1):"NAME SPIELER 2: "&P$(2):"NAME SPIE
LER 3: "&P$(3): :
119 DISPLAY AT(12,1):"STARTKAPITAL..: 50
00":"(DM 500-12000)": :"SPIELENDE BEI.:
1000000":"(>=10*KAPITAL)": :"KORREKTUR J
/N.: N"
120 DISPLAY AT(20,1):RPT$("=",28): :"(C)
 1984 BY HAGERA": :"STANDARTWERTE = <ENT
ER>"
121 ACCEPT AT(5,17)VALIDATE("23")SIZE(-1
):SPIELER
122 FOR Y1=1 TO SPIELER :: ACCEPT AT(7+Y
1,17)VALIDATE(UALPHA)SIZE(-5):P$(Y1):: I
F P$(Y1)="" THEN 122
123 NEXT Y1
124 ACCEPT AT(12,17)VALIDATE(DIGIT)SIZE(
-5):KAPITAL :: IF KAPITAL<500 OR KAPITAL
>12000 THEN 124
125 HV$=RPT$("3",11):: AR$=RPT$("4",11)
126 ACCEPT AT(15,17)VALIDATE(DIGIT)SIZE(
-7):SPIELENDE :: IF SPIELENDE<KAPITAL*10
 THEN 126
127 ACCEPT AT(18,17)VALIDATE("JN")SIZE(-
1):ACC$ :: IF ACC$="J" THEN 117
128 FOR Y1=1 TO 11 :: R(Y1)=40000 :: X(Y
1)=INT(Y1/2+.5)*100 :: V(Y1)=(ABS((Y1<2)
+(Y1<4)+(Y1<6)+(Y1<8)+(Y1<10))+1)*250
129 ST$(Y1)=RPT$("0"&STR$(X(Y1)),23):: N
OTE$(Y1)="BBR" :: NEXT Y1
130 FOR Y1=1 TO SPIELER :: K(Y1)=KAPITAL
 :: LK$(Y1)=RPT$("0",44):: NEXT Y1
131 P=1
132 COU=5
133 FOR Y1=1 TO 25 :: READ A$(Y1):: NEXT
 Y1
134 FOR Y1=1 TO 11 :: Z(Y1)=INT(Y1/2+.5)
*100 :: NEXT Y1
135 ! ****BRANCHENINDEX****
136 IF KEY1=48 THEN S(P)=S(P)+5
137 CALL CLEAR :: CALL CHCOL(13,2):: DIS
PLAY AT(1,1):"BRANCHENINDEX";P;P$(P)
138 DISPLAY AT(3,1):USING 111:"AG","BRIE
F","KURS","VORR."
139 FOR Y1=1 TO 11 :: DISPLAY AT(4+Y1,1)
:USING 111:A$(Y1),Z(Y1),X(Y1),V(Y1):: NE
XT Y1
140 DISPLAY AT(21,1):RPT$("*",28): :"SIE
 HABEN";COU;"HANDLUNGSPUNKTE":"BITTE AKT
IONSTASTE DRUECKEN!"
141 CALL KEY(0,KEY1,STA1):: IF STA1=0 TH
EN 141
142 IF KEY1<48 OR KEY1>57 THEN 357
143 ON KEY1-47 GOTO 135,144,151,171,190,
194,416,227,237,244
144 ! ****KAPITALLISTE****
145 IF KEY1=49 THEN S(P)=S(P)+5
```

```
146 CALL CLEAR :: CALL CHCOL(13,2):: DIS
PLAY AT(1,1):"KAPITALLISTE";P;P$(P)
147 DISPLAY AT(3,1):USING "UNVERBRIEFT D
M        ########":K(P)
148 DISPLAY AT(4,1):RPT$("*",28):"VERBRI
EFT:" :: DISPLAY AT(6,1):USING 111:"AG",
"UMSZ.","HABE","KURS"
149 DD=0 :: FOR Y1=1 TO 11 :: DD=DD+X(Y1
)*W(P,Y1):: DISPLAY AT(7+Y1,1):USING 111
:A$(Y1),NOTE$(Y1),W(P,Y1),X(Y1):: NEXT Y
1
150 DISPLAY AT(20,1):USING "GESAMTKAPITA
L DM    ##########":K(P)+DD :: GOTO 140
151 ! ****AKTIEN KAUFEN****
152 IF COU-2<0 THEN DISPLAY AT(24,1):"FA
LSCHE EINGABE!" :: CALL PAUSE(50):: GOTO
 140
153 CALL CLEAR :: CALL CHCOL(15,13)
154 DISPLAY AT(1,1):"ANKAUF VON AKTIEN:"
: :"SPIELER";P;P$(P):"UNVERBR. KAPITAL D
M:";K(P): :"WAS MOECHTEN SIE KAUFEN?"
155 DISPLAY AT(8,2):" <0> NICHTS
VORR KURS"
156 FOR Y1=1 TO 11 :: DISPLAY AT(9+Y1,2-
INT(Y1/10)):"<"&STR$(Y1)&"> "&A$(Y1):: D
ISPLAY AT(9+Y1,20):USING 113:V(Y1),X(Y1)
:: NEXT Y1
157 DISPLAY AT(22,1):"EINGABE DER NUMMER
:":RPT$("*",28):: ACCEPT AT(22,21)SIZE(2
)VALIDATE(DIGIT):AA :: IF AA=0 THEN 357
158 IF AA>11 OR AA=B THEN DISPLAY AT(24,
1)BEEP:"FALSCHE EINGABE!" :: GOTO 157 EL
SE CALL HCHAR(8,1,32,544)
159 DISPLAY AT(6,26):AA: :"AG          B
RIEF KURS VORR." :: DISPLAY AT(9,1):USIN
G 111:A$(AA),Z(AA),X(AA),V(AA)
160 DISPLAY AT(12,1):"MAXIMALE ANKAUFMEN
GE.:";MIN(MIN(9999,V(AA)),INT(K(P)/MAX(X
(AA),Z(AA))))
161 DISPLAY AT(14,1):"ANKAUFMENGE IN STU
ECK:" :: DISPLAY AT(22,1):RPT$("*",28)
162 ACCEPT AT(14,24)SIZE(4)VALIDATE(DIGI
T):BU :: IF BU<1 THEN NOTE$(AA)="BBR" EL
SE IF BU>V(AA)THEN NOTE$(AA)="BG" ELSE 1
64
163 DISPLAY AT(23,1):A$(13):: CALL PAUSE
(51):: GOTO 153
164 IF BU*MAX(X(AA),Z(AA))>K(P)THEN DISP
LAY AT(23,1):A$(14):: CALL PAUSE(50):: G
OTO 153
165 A=AA :: DD=MAX(X(A),Z(A)):: HH=MIN(X
(A),Z(A))
166 DISPLAY AT(17,1):USING "KAUFPREIS BA
R DM: ########":BU*HH :: DISPLAY AT(18,1)
:USING "MAKLERGEBUEHR DM: ########":INT(B
U*X(A)/50)
167 DISPLAY AT(19,1):USING "AUFPREIS AGI
O DM: ########":ABS(DD-HH)*BU :: DISPLAY
AT(20,1):USING "EFFEKTIVSUMME DM: ######
#":BU*HH+INT(BU*X(A)/50)+ABS(DD-HH)*BU
168 W(P,A)=W(P,A)+BU :: NOTE$(A)="GELD"
:: K(P)=K(P)-BU*DD :: V(A)=V(A)-BU :: IF
 K(P)<0 THEN S(P)=S(P)+ABS(K(P)):: K(P)=
0
169 LK$(P)=SEG$(LK$(P),1,4*A-4)&RPT$("0"
,4-LEN(STR$(DD)))&STR$(DD)&SEG$(LK$(P),4
*A+1,44-(4*A))
170 DISPLAY AT(23,1):"ANWEISUNG AUSGEFUE
HRT!" :: COU=COU-2 :: S(P)=S(P)+INT((BU*
X(A))/50):: CALL PAUSE(500):: GOTO 144
171 ! **AKTIEN VERKAUFEN**
172 IF COU-2<0 THEN 152
173 CALL CLEAR :: CALL CHCOL(15,7)
174 DISPLAY AT(1,1):"VERKAUF VON AKTIEN:
": :"SPIELER";P;P$(P):"UNVERBR. KAPITAL
DM:";K(P): :"WAS WOLLEN SIE VERKAUFEN?"
175 DISPLAY AT(8,2):" <0> NICHTS        H
ABEN KURS"
176 FOR Y1=1 TO 11 :: DISPLAY AT(9+Y1,2-
INT(Y1/10)):"<"&STR$(Y1)&"> "&A$(Y1):: D
ISPLAY AT(9+Y1,20):USING 113:W(P,Y1),X(Y
1):: NEXT Y1
177 DISPLAY AT(22,1):"EINGABE DER NUMMER
:":RPT$("*",28):: ACCEPT AT(22,21)SIZE(2
)VALIDATE(DIGIT):BB :: IF BB=0 THEN 357
178 IF BB>11 OR BB=A THEN DISPLAY AT(24,
1)BEEP:"FALSCHE EINGABE!" :: GOTO 177 EL
SE CALL HCHAR(8,1,32,544)
179 IF W(P,BB)=0 THEN DISPLAY AT(22,1):R
PT$("*",28):A$(15):: CALL PAUSE(50):: GO
TO 173
180 DISPLAY AT(6,26):BB: :"AG          B
RIEF KURS  HABE" :: DISPLAY AT(9,1):USIN
G 111:A$(BB),Z(BB),X(BB),W(P,BB)
181 HH=VAL(SEG$(LK$(P),BB*4-3,4)):: DISP
LAY AT(12,1):USING "LETZTER KAUF  BAR DM
: ####":HH
182 DISPLAY AT(13,1):USING "EFFEKTIVKAUF
PREIS DM: ####":HH+INT(HH/50)+INT(X(BB)/
50)
183 DISPLAY AT(15,1):"VERKAUFSMENGE STUE
CK:" :: DISPLAY AT(22,1):RPT$("*",28)
184 ACCEPT AT(15,24)SIZE(4)VALIDATE(DIGI
T):SA :: IF SA<1 OR SA>W(P,BB)THEN DISPL
AY AT(23,1):A$(16):: CALL PAUSE(50):: GO
TO 173
185 IF SA*X(BB)<1 THEN DISPLAY AT(23,1):
A$(20):: GOTO 173
186 B=BB :: W(P,B)=W(P,B)-SA :: NOTE$(B)
="BRIEF" :: K(P)=K(P)+SA*X(B):: V(B)=V(B
)+SA
187 DISPLAY AT(17,1):USING "VERKAUFSWERT
   DM: ########":X(B)*SA :: DISPLAY AT(18
,1):USING "MAKLERGEBUEHR  DM: ########":I
NT(X(B)*SA/50)
188 DISPLAY AT(19,1):USING "REINERLOES B
AR DM: ########":X(B)*SA-INT(X(B)*SA/50)
189 DISPLAY AT(23,1):"ANWEISUNG AUSGEFUE
HRT!" :: COU=COU-2 :: S(P)=S(P)+INT((SA*
X(B))/50):: CALL PAUSE(300):: GOTO 144
190 ! ****BOERSENAKTION****
191 IF COU-3<0 THEN DISPLAY AT(24,1):"FA
LSCHE EINGABE!" :: CALL PAUSE(50):: GOTO
 140
192 DISPLAY AT(1,1)ERASE ALL BEEP:"BOERS
ENAKTION",P;P$(P): :A$(25): :RPT$("*",28
):: ACCEPT AT(3,28)SIZE(1)VALIDATE("JN")
:ACC$
193 IF ACC$="N" THEN 357 ELSE R1=INT(RND
*13)+1 :: ON R1 GOTO 276,301,301,301,301
,326,301,301,301,301,331,341,349
194 ! ****MANAGEMENT****
195 IF COU-1<0 THEN 357 ELSE S(P)=S(P)+1
0
196 CALL CLEAR :: CALL CHCOL(7,12)
197 DISPLAY AT(1,1):"MANAGEMENT":RPT$("*
",28): :"BITTE WAEHLEN SIE:"
198 DISPLAY AT(6,1):"<0> MENUE":"<1> INF
ORMATION":"<2> HAUPTVERSAMMLUNG"
199 CALL KEY(0,KEY3,STA3):: IF STA3=0 OR
 KEY3<48 OR KEY3>50 THEN 199
200 IF KEY3=48 THEN COU=COU-1
201 ON KEY3-47 GOTO 357,202,215
202 ! **INFORMATION**
203 DISPLAY AT(1,1)ERASE ALL:"INFORMATIO
N MANAGEMENT":RPT$("*",28): :"AUSKUNFT F
UER: "&P$(P):: GOSUB 363
204 DISPLAY AT(24,1):"AG-NUMMER EINGEBEN
:" :: ACCEPT AT(24,22)SIZE(2)VALIDATE(NU
MERIC):MI :: IF MI>11 THEN 204
205 CALL HCHAR(6,1,32,576):: S(P)=S(P)+5
206 DISPLAY AT(6,1):"AG: "&A$(MI): :RPT$
("*",28)
207 DISPLAY AT(10,1):"STAMMKAPITAL:";(V(
MI)+W(1,MI)+W(2,MI)+W(3,MI))*Z(MI)
208 DISPLAY AT(11,1):"RUECKLAGEN...:";R(M
I)
209 DISPLAY AT(12,1):"VORSITZENDER: ";P$
(VAL(SEG$(AR$,MI,1)))
210 DISPLAY AT(14,1):"KURSENTWICKLUNG:"
211 FOR I=3 TO 21 STEP 3 :: LI=I/3+15
212 DISPLAY AT(LI,1):USING "##. ####  ##
. ####  ##. ####":I,VAL(SEG$(ST$(MI),I*4
-3,4)),I+1,VAL(SEG$(ST$(MI),(I+1)*4-3,4)
),I+2,VAL(SEG$(ST$(MI),(I+2)*4-3,4))
213 NEXT I :: DISPLAY AT(24,1):"<ENTER>
DRUECKEN"
```

```
214 CALL KEY(0,KEY3,STA3):: IF STA3=0 TH
EN 214 ELSE 196
215 ! **HAUPTVERSAMMLUNG**
216 DISPLAY AT(1,1)ERASE ALL:"HAUPTVERSA
MMLUNG: "&P$(P):RPT$("*",28): :"NR. AG-N
AME      % è VORSITZ"
217 FOR Y1=1 TO 11
218 DISPLAY AT(Y1+6,1):USING 112:Y1,A$(Y
1),INT(100/(V(Y1)+W(1,Y1)+W(2,Y1)+W(3,Y1
))*W(P,Y1)),SEG$(HV$,Y1,1),P$(VAL(SEG$(A
R$,Y1,1)))
219 NEXT Y1
220 DISPLAY AT(19,1):"BESTIMMUNG UEBER D
IVIDENDEN:":"<1> UNBEDINGT NEIN":"<2> JU
NGE AKTIEN ANSTATT":"<3> NORMALISIEREN"
221 DISPLAY AT(24,1):"BITTE JEWEILS EING
EBEN!"
222 FOR Y1=1 TO 11
223 IF W(P,Y1)<INT((V(Y1)+W(1,Y1)+W(2,Y1
)+W(3,Y1))/4)THEN 225
224 ACCEPT AT(Y1+6,20)SIZE(-1)VALIDATE("
123"):BD :: HV$=SEG$(HV$,1,Y1-1)&STR$(BD
)&SEG$(HV$,Y1+1,12-Y1)
225 NEXT Y1
226 DISPLAY AT(24,1):"TASTE DRUECKEN" ::
 CALL PAUSE(800):: GOTO 196
227 ! ****KAPITALVGL****
228 IF KEY1=55 THEN S(P)=S(P)+5
229 CALL CLEAR :: CALL CHCOL(13,2):: DIS
PLAY AT(1,1):"KAPITALVERGLEICH";P;P$(P)
230 FOR Y1=1 TO SPIELER :: HG=0 :: FOR Y
2=1 TO 11 :: HG=HG+X(Y2)*W(Y1,Y2):: NEXT
 Y2
231 DISPLAY AT(6*Y1-3,3):USING "###### B
AR DM ########":P$(Y1),K(Y1)
232 DISPLAY AT(6*Y1-2,1):USING "+ VERBRI
EFT  DM ########":HG
233 DISPLAY AT(6*Y1-1,1):USING "= BRUTTO
     DM ########":K(Y1)+HG
234 DISPLAY AT(6*Y1,1):USING "- SCHULDEN
   DM ########":S(Y1)
235 DISPLAY AT(6*Y1+1,1):USING "= NETTO
     DM ########":K(Y1)+HG-S(Y1)
236 NEXT Y1 :: IF ENDE=1 THEN 438 ELSE 1
40
237 ! ****INFORMATION****
238 IF KEY1=56 THEN S(P)=S(P)+5
239 CALL CLEAR :: CALL CHCOL(13,2)
240 DISPLAY AT(1,1):"AKTIENVERGLEICH";P;
P$(P):: DISPLAY AT(3,1):USING "AG
  ##### ##### #####":P$(1),P$(2),P$(3)
241 FOR Y1=1 TO 11 :: IF SPIELER=3 THEN
DISPLAY AT(3+Y1,1):USING "########## ##
## ####  ####":A$(Y1),W(1,Y1),W(2,Y1),W
(3,Y1)
242 IF SPIELER=2 THEN DISPLAY AT(3+Y1,1)
:USING "########## #### ####":A$(Y1),W(1
,Y1),W(2,Y1)
243 NEXT Y1 :: GOTO 140
244 ! ***GRAPHIK-SUBRTNS***
245 IF KEY1=57 THEN S(P)=S(P)+10
246 CALL CLEAR :: CALL CHCOL(2,16):: DIS
PLAY AT(1,8):"AUSWERTUNG "&P$(P)
247 PWI,PWII=0
248 DISPLAY AT(2,1):"OVER":"2000":"1750"
:"1500":"1250":"1000":" 900":" 800":" 70
0":" 600":" 500":" 400":" 350":" 300"
249 DISPLAY AT(16,1):" 250":" 200":" 150
":" 100":"  50":"FLOW"
250 CALL VCHAR(1,7,124,22):: CALL HCHAR(
22,3,61,28)
251 DISPLAY AT(3,8):"BITTE WAEHLEN SIE:"
252 FOR I=1 TO 11 :: DISPLAY AT(I+4,8):U
SING "############## ##":A$(I),I :: NEXT
I
253 DISPLAY AT(16,8):"MENUE / EINZEL 0"
:: DISPLAY AT(18,8):"NUMMER 1-12.:" :: D
ISPLAY AT(19,8):"NUMMER 1-11.:"
254 FOR I=1 TO 2
255 ACCEPT AT(I+17,22)SIZE(2)VALIDATE(NU
MERIC):GR(I):: IF GR(I)>11 THEN 255
256 IF GR(1)=0 THEN 357
257 IF GR(2)=0 THEN II=1 :: GR(2)=22 ELS
E II=2
258 NEXT I
259 FOR I=2 TO 19 :: CALL HCHAR(I,8,32,2
4):: NEXT I
260 DISPLAY AT(23,1):RPT$(CHR$(130),4)&"
 "&A$(GR(1)):RPT$(CHR$(137),4)&" "&A$(GR
(2))
261 FOR Y1=1 TO II
262 PWI=VAL(SEG$(ST$(GR(Y1)),1,4))
263 PWI=20-(INT(PWI/50+1)*-(PWI<401)+INT
(PWI/100+5)*(PWI<1001)*(PWI>400)+INT(PWI
/250+11)*(PWI<2001)*(PWI>1000)-(PWI>2000
)*20)
264 CALL HCHAR(PWI+2,8,123+Y1*7)
265 FOR Y2=2 TO 23
266 PWII=VAL(SEG$(ST$(GR(Y1)),Y2*4-3,4))
267 PWII=20-(INT(PWII/50+1)*-(PWII<401)+
INT(PWII/100+5)*(PWII<1001)*(PWII>400)+I
NT(PWII/250+11)*(PWII<2001)*(PWII>1000)-
(PWII>2000)*20)
268 CALL HCHAR(PWII+2,7+Y2,123+Y1*7):: I
F PWI=PWII THEN 272
269 PW3=PW3+1 :: IF PWII>PWI THEN CALL H
CHAR(PWII-PW3+3,Y2+7,123+Y1*7):: PWI=PWI
+1 :: GOTO 269
270 PW3=0
271 PW3=PW3+1 :: IF PWII<PWI THEN CALL H
CHAR(PWII+PW3+1,Y2+7,123+Y1*7):: PWI=PWI
-1 :: GOTO 271
272 PWI=PWII :: PW3=0 :: NEXT Y2 :: NEXT
 Y1
273 DISPLAY AT(23,20):"<ENTER>" :: DISPL
AY AT(24,19):"DRUECKEN!"
274 CALL KEY(0,KEY3,STA3):: IF KEY3<>13
THEN 274
275 FOR I=2 TO 21 :: CALL HCHAR(I,8,32,2
4):: NEXT I :: DISPLAY AT(23,1):RPT$(" "
,56):: GOTO 251
276 ! ***AKTIONEN***
277 ! *DIVIDENDE*
278 IF SEG$(HV$,R1,1)="1" THEN DISPLAY A
T(24,1):A$(24):: CALL PAUSE(150):: GOTO
192
279 CALL CHCOL(8,16)
280 R1=INT(RND*13):: IF R1=0 OR R1=12 TH
EN DISPLAY AT(6,1):"DIVIDENDE / AUFWERTU
NG"
281 IF R1=0 OR R1=12 THEN GOSUB 363 :: D
ISPLAY AT(24,1):"AG-NUMMER DIV./AUFWERTU
NG:" :: ACCEPT AT(24,27)SIZE(2)VALIDATE(
DIGIT):R1 :: IF R1<1 OR R1>11 THEN 280
282 CALL CLEAR :: DISPLAY AT(1,1):"BOERS
ENAKTION";P;P$(P):RPT$("*",28)
283 IF X(R1)<Z(R1)THEN 291
284 DISPLAY AT(6,1):"DIVIDENDE 4%": :"AG
          DIFF: ALT: NEU:"
285 DISPLAY AT(9,1):USING 111:A$(R1),INT
(X(R1)/25),X(R1),X(R1)-INT(X(R1)/25)
286 DISPLAY AT(11,1):"SPIELER       HABEN
  xDM =GEW"
287 FOR Y1=1 TO SPIELER :: DISPLAY AT(11
+Y1,1):USING "1 ######       #### #### ##
##":P$(Y1),W(Y1,R1),INT(X(R1)/25),W(Y1,R
1)*INT(X(R1)/25):: NEXT Y1
288 DISPLAY AT(23,1):"AKZEPTIEREN SIE <J
/N>?" :: ACCEPT AT(23,24)SIZE(1)VALIDATE
("JN"):ACC$ :: IF ACC$="N" THEN 290
289 FOR Y1=1 TO SPIELER :: K(Y1)=K(Y1)+W
(Y1,R1)*INT(X(R1)/25):: NEXT Y1 :: X(R1)
=X(R1)-INT(X(R1)/25)
290 COU=COU-3 :: GOTO 140
291 CALL PAUSE(35):: DISPLAY AT(6,1):"AU
FWERTUNG (KURS=BRIEF)": :"AG            A
UFW: ALT: NEU:"
292 DISPLAY AT(9,1):USING 111:A$(R1),Z(R
1)-X(R1),X(R1),Z(R1)
293 DISPLAY AT(11,1):"SPIELER       HABEN
  xDM =DM-"
294 FOR Y1=1 TO SPIELER :: DISPLAY AT(11
+Y1,1):USING "# #####       #### #### ##
##":Y1,P$(Y1),W(Y1,R1),Z(R1)-X(R1),W(Y1,
R1)*(Z(R1)-X(R1))
295 IF K(Y1)<(W(Y1,R1)*(Z(R1)-X(R1)))THE
N K(Y1)=K(Y1)-(W(Y1,R1)*(Z(R1)-X(R1))):
 GOTO 299
```

```
296 DISPLAY AT(17,1):"ACHTUNG, "&P$(Y1)&
"!":"AUFWERTUNG...":"DER AUSGLEICH ERFOL
GT DURCH VERLUST IHRER":A$(R1)&"-AKTIEN.
"
297 K(Y1)=K(Y1)+W(Y1,R1)*X(R1):: V(R1)=V
(R1)+W(Y1,R1):: W(Y1,R1)=0
298 CALL PAUSE(50)
299 NEXT Y1 :: X(R1)=Z(R1):: DISPLAY AT(
23,1):"AUTOMATISCHE DURCHFUEHRUNG!"
300 COU=COU-3 :: CALL PAUSE(51):: GOTO 1
40
301 ! HAUSSE-BAISSE
302 CALL CHCOL(11,5)
303 IF R1=2 OR R1=7 THEN UP=1 :: DW=1 EL
SE IF R1=3 OR R1=8 THEN UP=1 :: DW=2 ELS
E IF R1=4 OR R1=9 THEN UP=2 :: DW=1 ELSE
 UP=2 :: DW=2
304 R2=INT(RND*13):: IF R2=0 OR R2=12 TH
EN GOSUB 363
305 IF R1>6 THEN 316
306 IF R2<>0 AND R2<>12 THEN 309
307 DISPLAY AT(23,1):"AG-NUMMER HAUSSE:"
 :: ACCEPT AT(23,19)SIZE(2)VALIDATE(DIGI
T):R2 :: IF R2<1 OR R2>11 THEN 307
308 CALL HCHAR(8,1,32,416)
309 DISPLAY AT(6,1):"AG                DIFF:
ALT: NEU:":"HAUSSE:" :: R1=INT(RND*5)+1
:: DISPLAY AT(8,1):USING 111:A$(R2),R1*U
P,X(R2),X(R2)+R1*UP
310 DISPLAY AT(10,1):"BAISSE:" :: FOR Y1
=1 TO 11 :: IF Y1=R2 THEN 312
311 DISPLAY AT(10+Y1,1):USING 111:A$(Y1)
,-R1*DW,X(Y1),X(Y1)-R1*DW
312 NEXT Y1
313 DISPLAY AT(24,1):"AKZEPTIEREN SIE <J
/N>?" :: ACCEPT AT(24,24)SIZE(1)VALIDATE
("JN"):ACC$ :: IF ACC$="N" THEN COU=COU-
3 :: GOTO 140
314 FOR Y1=1 TO 11 :: IF Y1=R2 THEN X(Y1
)=X(Y1)+R1*UP ELSE X(Y1)=X(Y1)-R1*DW
315 NEXT Y1 :: DISPLAY AT(24,1):"DURCHGE
FUEHRT!" :: CALL PAUSE(30):: COU=COU-1 :
: GOTO 135
316 IF R2<>0 AND R2<>12 THEN 319
317 DISPLAY AT(23,1):"AG-NUMMER BAISSE:"
 :: ACCEPT AT(23,19)SIZE(2)VALIDATE(DIGI
T):R2 :: IF R2<1 OR R2>11 THEN 317
318 CALL HCHAR(8,1,32,416)
319 DISPLAY AT(6,1):"AG                DIFF:
ALT: NEU:":"BAISSE:" :: R1=INT(RND*5)+1
:: DISPLAY AT(8,1):USING 111:A$(R2),-R1*
DW,X(R2),X(R2)-R1*DW
320 DISPLAY AT(10,1):"HAUSSE:" :: FOR Y1
=1 TO 11 :: IF Y1=R2 THEN 322
321 DISPLAY AT(10+Y1,1):USING 111:A$(Y1)
,R1*UP,X(Y1),X(Y1)+R1*UP
322 NEXT Y1
323 DISPLAY AT(24,1):"AKZEPTIEREN SIE <J
/N>?" :: ACCEPT AT(24,24)SIZE(1)VALIDATE
("JN"):ACC$ :: IF ACC$="N" THEN COU=COU-
3 :: GOTO 140
324 FOR Y1=1 TO 11 :: IF Y1=R2 THEN X(Y1
)=X(Y1)-R1*DW ELSE X(Y1)=X(Y1)+R1*UP
325 NEXT Y1 :: DISPLAY AT(24,1):"DURCHGE
FUEHRT!" :: CALL PAUSE(30):: COU=COU-1 :
: GOTO 135
326 !MULTIPLE ACTION FLOOD
327 CALL CHCOL(2,7)
328 DISPLAY AT(6,1):"MULTIPLE ACTION FLO
OD": :"AG                DIFF% ALT: NEU:"
329 FOR Y1=1 TO 11 :: R1=INT(RND*151)-75
 :: DISPLAY AT(9+Y1,1):USING 111:A$(Y1),
R1,X(Y1),X(Y1)+INT(X(Y1)*R1/100):: X(Y1)
=X(Y1)+INT(X(Y1)*R1/100):: CALL PAUSE(51
):: NEXT Y1
330 DISPLAY AT(24,1):"AUTOMATISCHE DURCH
FUEHRUNG!" :: COU=COU-3 :: FOR I=110 TO
990 STEP 10 :: CALL SOUND(-20,I,0):: NEX
T I :: CALL PAUSE(600):: GOTO 135
331 !MANUAL I
332 CALL CHCOL(14,15)
333 DISPLAY AT(6,1):"BROKERS MANUAL ACTI
VITY I" :: GOSUB 363
334 DISPLAY AT(23,1):"AG-NUMMER STEIGT:"
:"AG-NUMMER FAELLT:" :: ACCEPT AT(23,19)
SIZE(2)VALIDATE(DIGIT):R1 :: IF R1<1 OR
R1>11 THEN 334
335 ACCEPT AT(24,19)SIZE(2)VALIDATE(DIGI
T):R2 :: IF R2<1 OR R2>11 THEN 335
336 CALL HCHAR(7,1,32,576):: R3=INT(RND*
50)+1 :: DISPLAY AT(8,1):"ES STEIGT:":"A
G              DIFF% ALT: NEU:"
337 DISPLAY AT(10,1):USING 111:A$(R1),R3
,X(R1),INT(X(R1)+X(R1)*R3/100):: X(R1)=I
NT(X(R1)+X(R1)*R3/100)
338 DISPLAY AT(12,1):"ES FAELLT:":"AG
          DIFF% ALT: NEU:"
339 DISPLAY AT(14,1):USING 111:A$(R2),-R
3,X(R2),INT(X(R2)-X(R2)*R3/100):: X(R2)=
INT(X(R2)-X(R2)*R3/100):: CALL PAUSE(40)
340 DISPLAY AT(24,1):"AUTOMATISCHE DURCH
FUEHRUNG!" :: CALL PAUSE(200):: COU=COU-
3 :: GOTO 135
341 !MANUAL II
342 CALL CHCOL(14,15)
343 DISPLAY AT(6,1):"BROKERS MANUAL ACTI
VITY II"
344 DISPLAY AT(8,1):"<1>  DIVIDENDE / AU
FWERTUNG <2>  HAUSSE-BAISSE ]¿  [è   <3>
  HAUSSE-BAISSE ]¿  [èè  <4>  HAUSSE-BAI
SSE ]¿¿  [è     "
345 DISPLAY AT(12,1):"<5>  HAUSSE-BAISSE
 ]¿¿  [èè  <6>  MULTIPLE ACTION FLOOD  <7
>  HAUSSE-BAISSE [¿  ]è   <8>  HAUSSE-BA
ISSE [¿  ]èè    "
346 DISPLAY AT(16,1):"<9>  HAUSSE-BAISSE
 [¿¿  ]è   <10> HAUSSE-BAISSE [¿¿  ]èè  <1
1> MANUAL ACTIVITY I       <12> FINANZAMT
"
347 DISPLAY AT(23,1):"BOERSENAKTION-NUMM
ER:" :: ACCEPT AT(23,23)SIZE(2)VALIDATE(
DIGIT):R1 :: IF R1<1 OR R1>12 THEN 347
348 CALL HCHAR(8,1,32,416):: ON R1 GOTO
276,301,301,301,301,326,301,301,301,301,
331,349
349 !FISKUS
350 CALL CHCOL(2,13)
351 DISPLAY AT(6,1):"DER FISKUS BITTET Z
UR KASSE:": :"KAPITALERTRAGSSTEUER":"(25
% DES BARKAPITALS, SOFERNDIESES GROESSER
 IST ALS DAS 1.5FACHE STARTKAPITAL)."
352 DISPLAY AT(13,1):USING "STARTKAPITAL
 DM:    #####":KAPITAL :: DISPLAY AT(14,
1):USING "FREIBETRAG   DM:    #####":INT
(KAPITAL*1.5)
353 DISPLAY AT(16,1):"NAME: BAR DM: STNE
TTO STEUER"
354 FOR Y1=1 TO SPIELER :: DISPLAY AT(16
+Y1,1):USING "##### ######## ######## ####
##":P$(Y1),K(Y1),MAX(0,K(Y1)-INT(KAPITAL
*1.5)),INT(MAX(0,K(Y1)-INT(KAPITAL*1.5))
/4)
355 IF K(Y1)>1.5*KAPITAL THEN K(Y1)=K(Y1
)-INT(MAX(0,K(Y1)-(KAPITAL*1.5)/4))
356 NEXT Y1 :: DISPLAY AT(24,1):A$(21)::
 CALL PAUSE(350):: GOTO 227
357 ! *MENUE*
358 CALL CLEAR :: CALL CHCOL(7,2):: DISP
LAY AT(1,1):"MENUE",P;P$(P):RPT$("*",28)
359 DISPLAY AT(4,1):"KEY FUNKTION
   HP  DM":RPT$("=",28)
360 DISPLAY AT(6,1):"<0> BRANCHENINDEX
    0   5":"<1> KAPITALLISTE       0   5
":"<2> AKTIEN KAUFEN       2  2%":"<3> AK
TIEN VERKAUFEN   2  2%"
361 DISPLAY AT(10,1):"<4> BOERSENAKTIONE
N  1-3   0":"<5> MANAGEMENT            1  1
0":"<6> SPIELERWECHSEL     0   0":"<7> K
APITALVERGLEICH   0   5"
362 DISPLAY AT(14,1):"<8> AKTIENVERGLEIC
H    0   5":"<9> GRAPHIKAUSWERTUNG  0  1
0": :"< > MENUE" :: GOTO 140
363 ! ****WAHLPROGRAMM****
364 DISPLAY AT(8,1):"NR.: AG
HABE KURS" :: FOR I=1 TO 11 :: DISPLAY A
T(I+8,2-INT(I/10)):"<"&STR$(I)&"> "&A$(I
)
```

```
365 DISPLAY AT(I+8,20):USING 113:W(P,I),
X(I):: NEXT I :: RETURN
366 ! *TITEL UND WAHL*
367 CALL CLEAR :: DISPLAY AT(1,4):"B  R
 O  K  E  R   III":RPT$("=",28)
368 DISPLAY AT(12,5):RPT$("=",28): :"BIT
TE WAEHLEN SIE:": :"<1> NEUES SPIEL":"<2
> ALTES SPIEL"
369 CALL KEY(0,KEY3,STA3):: IF STA3=0 OR
 KEY3<49 OR KEY3>50 THEN 369
370 ON KEY3-48 GOTO 117,371
371 ! *DATEN LESEN*
372 CALL HCHAR(19,1,32,192)
373 DISPLAY AT(19,1):"DATENDISKETTE EINL
EGEN!": :"DISKETTENNAME:": :"DATEINAME..
..:"
374 ACCEPT AT(21,16)SIZE(10):DIN$
375 ACCEPT AT(23,16)SIZE(20):DAN$
376 ON ERROR 117
377 OPEN #1:"DSK."&DIN$&"."&DAN$,INTERNA
L,INPUT ,VARIABLE 164
378 INPUT #1:SPIELER,KAPITAL,SPIELENDE,P
379 INPUT #1:P$(1),P$(2),P$(3)
380 FOR I=1 TO 10 :: INPUT #1:R(I),:: NE
XT I :: INPUT #1:R(I)
381 FOR I=1 TO 10 :: INPUT #1:X(I),:: NE
XT I :: INPUT #1:X(I)
382 FOR I=1 TO 10 :: INPUT #1:V(I),:: NE
XT I :: INPUT #1:V(I)
383 FOR J=1 TO 3 :: FOR I=1 TO 11 :: INP
UT #1:W(J,I),:: NEXT I :: INPUT #1:S(J):
: NEXT J
384 FOR I=1 TO 3 :: INPUT #1:LK$(I):: NE
XT I
385 INPUT #1:K(1),K(2),K(3)
386 INPUT #1:HV$,AR$
387 FOR I=1 TO 10 :: INPUT #1:NOTE$(I),:
: NEXT I :: INPUT #1:NOTE$(I)
388 FOR I=1 TO 11 :: INPUT #1:ST$(I):: N
EXT I
389 CLOSE #1
390 CALL HCHAR(19,1,32,196):: DISPLAY AT
(20,1):"PROGRAMMDISKETTE EINLEGEN   DANN
 <ENTER> DRUECKEN."
391 CALL KEY(0,KEY3,STA3):: IF STA3=0 OR
 KEY3<>13 THEN 391
392 GOTO 132
393 ! ** DATEN SAVEN **
394 CALL CLEAR :: DISPLAY AT(1,1):"BROKE
R III":RPT$("*",28): :"> INITIALISIERTE"
:"  DISKETTE EINLEGEN": : :"DANN BITTE E
INGEBEN:"
395 DISPLAY AT(12,1):"DISKETTENNAME:":"D
ATEINAME....:"
396 ACCEPT AT(12,16)SIZE(10):DIN$
397 ACCEPT AT(13,16)SIZE(10):DAN$
398 ON ERROR 396 :: IF DIN$=DAN$ THEN 41
1
399 OPEN #1:"DSK."&DIN$&"."&DAN$,INTERNA
L,OUTPUT,VARIABLE 164
400 PRINT #1:SPIELER,KAPITAL,SPIELENDE,P
401 PRINT #1:P$(1),P$(2),P$(3)
402 FOR I=1 TO 10 :: PRINT #1:R(I),:: NE
XT I :: PRINT #1:R(I)
403 FOR I=1 TO 10 :: PRINT #1:X(I),:: NE
XT I :: PRINT #1:X(I)
404 FOR I=1 TO 10 :: PRINT #1:V(I),:: NE
XT I :: PRINT #1:V(I)
405 FOR J=1 TO 3 :: FOR I=1 TO 11 :: PRI
NT #1:W(J,I),:: NEXT I :: PRINT #1:S(J):
: NEXT J
406 FOR I=1 TO 3 :: PRINT #1:LK$(I):: NE
XT I
407 PRINT #1:K(1),K(2),K(3)
408 PRINT #1:HV$,AR$
409 FOR I=1 TO 10 :: PRINT #1:NOTE$(I),:
: NEXT I :: PRINT #1:NOTE$(I)
410 FOR I=1 TO 11 :: PRINT #1:ST$(I):: N
EXT I
411 CLOSE #1 :: IF DIN$=DAN$ THEN 396
412 DISPLAY AT(20,1):"DATEN ABGESPEICHER
T!": :"PROGRAMMDISKETTE EINLEGEN.":"
DANN <ENTER> DRUECKEN."
413 CALL KEY(0,KEY3,STA3):: IF STA3=0 OR
 KEY3<>13 THEN 413
414 ON ERROR 413
415 RUN "DSK1.BROKER-1"
416 ! **SPIELERWECHSEL**
417 CALL CLEAR :: CALL CHCOL(5,11)
418 DISPLAY AT(1,1):"SPIELERWECHSEL";P;P
$(P):RPT$("*",28): :A$(25)
419 DISPLAY AT(6,1):"<N> NEIN":"<@> SPIE
LUNTERBRECHUNG":"< > BESTAETIGUNG": :RPT
$("*",28)
420 CALL KEY(0,KEY3,STA3):: IF STA3=0 TH
EN 420
421 IF KEY3=78 THEN 140
422 A,AA,B,BB=0 :: COU=5 :: DISPLAY AT(1
2,1):"BITTE WARTEN - ICH RECHNE!"
423 IF S(P)>0 AND K(P)>S(P)THEN GOSUB 48
7 ELSE IF S(P)>0 AND K(P)<S(P)THEN GOSUB
 491
424 HH=0 :: FOR Y1=1 TO 11 :: HH=HH+W(P,
Y1)*X(Y1):: NEXT Y1
425 IF (K(P)>=SPIELENDE AND S(P)=0)OR K(
P)+HH-S(P)<100 THEN ENDE=1
426 ! **NACHRICHTEN**
427 FOR Y1=1 TO 11 :: H$="0000"&STR$(X(Y
1)):: ST$(Y1)=SEG$(ST$(Y1),5,92)&SEG$(H$
,LEN(H$)-3,LEN(H$))
428 IF X(Y1)<50 THEN GOSUB 446 ELSE IF X
(Y1)>2000 THEN ON VAL(SEG$(HV$,Y1,1))GOS
UB 453,459,453
429 AR=VAL(SEG$(AR$,Y1,1)):: IF AR<4 AND
 X(Y1)>Z(Y1)THEN GOSUB 473
430 IF VAL(SEG$(AR$,Y1,1))<>P AND W(P,Y1
)>(V(Y1)+W(1,Y1)+W(2,Y1)+W(3,Y1)-W(P,Y1)
)THEN GOSUB 479
431 IF V(Y1)=0 AND NOTE$(Y1)="GELD" THEN
 NOTE$(Y1)="BZ"
432 IF VAL(SEG$(AR$,Y1,1))=P AND W(P,Y1)
<=(V(Y1)+W(1,Y1)+W(2,Y1)+W(3,Y1)-W(P,Y1)
)THEN GOSUB 483
433 NEXT Y1 :: P=P+1 :: IF P>SPIELER THE
N P=1
434 IF ENDE=1 THEN 436
435 IF KEY3=64 THEN 393 ELSE 357
436 ! **SPIELENDE**
437 CALL CLEAR :: CALL CHCOL(9,16):: DIS
PLAY AT(1,1):"SCHLUSSBERECHNUNG:" :: GOT
O 230
438 DISPLAY AT(23,1):"SPIEL BEENDET.":"D
RUECKEN SIE EINE TASTE!"
439 RESTORE 501
440 READ DA,F1,F2,F3
441 IF DA=0 THEN 439
442 CALL SOUND(DA,F1,0,F2,0,F3,0)
443 CALL KEY(0,KEY3,STA3):: IF STA3=0 TH
EN 440
444 ON ERROR 443
445 RUN "DSK1.BROKER-1"
446 ! **AUFWERTUNG**
447 DISPLAY AT(15,1):A$(17)&" "&A$(Y1)
448 DISPLAY AT(18,1):"AUFWERT. JE AKTIE
DM:"&STR$(Z(Y1)-X(Y1))
449 FOR Y2=1 TO SPIELER
450 IF K(Y2)<W(Y2,Y1)*(Z(Y1)-X(Y1))THEN
DISPLAY AT(19+Y2,1):P$(Y2)&" -AKTIENVERL
UST-" :: K(Y2)=K(Y2)+W(Y2,Y1)*X(Y1):: V(
Y1)=V(Y1)+W(Y2,Y1):: W(Y2,Y1)=0 :: GOTO
452
451 DISPLAY AT(19+Y2,1):P$(Y2)&" ZAHLT D
M: "&STR$(W(Y2,Y1)*(Z(Y1)-X(Y1)))&" AKTI
ENVERLUST-" :: K(Y2)=K(Y2)-W(Y2,Y1)*(Z(Y1
)-X(Y1))
452 NEXT Y2 :: X(Y1)=Z(Y1):: CALL PAUSE(
150):: RETURN
453 ! **DIVIDENDE**
454 DISPLAY AT(15,1):A$(18)&" "&A$(Y1)
455 DISPLAY AT(18,1):"AUSZAHL. JE AKTIE
DM: "&STR$(X(Y1)-2000)
456 FOR Y2=1 TO SPIELER
457 DISPLAY AT(19+Y2,1):P$(Y2)&" ERHAELT
: "&STR$(W(Y2,Y1)*(X(Y1)-2000))
458 NEXT Y2 :: X(Y1)=2000 :: CALL PAUSE(
150):: RETURN
```

```
459 ! **RUECKLAGEN**
460 DISPLAY AT(15,1):A$(21)&" "&A$(Y1)
461 DISPLAY AT(18,1):"RUECKLAGE BISHER:
"&STR$(R(Y1)):"UEBERSCHUESSE...: "&STR$(
X(Y1)-2000)
462 VG=V(Y1)+W(1,Y1)+W(2,Y1)+W(3,Y1):: D
ISPLAY AT(20,1):"RUECKLAGE JETZT.: "&STR
$(R(Y1)+(X(Y1)-2000)*VG)
463 R(Y1)=R(Y1)+(X(Y1)-2000)*VG
464 X(Y1)=2000
465 CALL PAUSE(150):: IF R(Y1)<INT((VG*Z
(Y1))/5+40000)THEN 472
466 V(Y1)=INT(V(Y1)*1.2):: FOR Y2=1 TO 3
 :: W(Y2,Y1)=INT(W(Y2,Y1)*1.2):: NEXT Y2
467 R(Y1)=R(Y1)-INT(VG*Z(Y1)/5)
468 DISPLAY AT(15,1):A$(22):A$(Y1)
469 FOR Y2=1 TO SPIELER
470 DISPLAY AT(19+Y2,1):P$(P)&":"&STR$(I
NT(W(Y2,Y1)/2))&"JUNGE AKTIEN."
471 NEXT Y2 :: CALL PAUSE(150)
472 RETURN
473 ! **AR-LOHN**
474 IF NOTE$(Y1)="BRIEF" OR NOTE$(Y1)="B
BR" THEN IF RND*2<1 THEN 478
475 DISPLAY AT(15,1):"DER AR-VORSITZENDE
 DER AG":A$(Y1)&", "&P$(SEG$(AR$,Y1,1))&
" ERHAELT":"DM...:";35*Y1
476 K(VAL(SEG$(AR$,Y1,1)))=K(VAL(SEG$(AR
$,Y1,1)))+35*Y1
477 CALL PAUSE(150)
478 RETURN
479 ! **AR-WAHL**
480 DISPLAY AT(15,1):"ZUM AR-VORSITZENDE
N DER AG":A$(Y1)&" WURDE GEWAEHLT:":P$(P
)
481 AR$=SEG$(AR$,1,Y1-1)&STR$(P)&SEG$(AR
$,Y1+1,11-Y1)
482 CALL PAUSE(150):: RETURN
483 ! **AR-RUECKTRITT**
484 DISPLAY AT(15,1):" VOM AR-VORSITZ DE
R AG":A$(Y1)&" IST":"ZURUECKGETRETEN: "&
P$(P)
485 AR$=SEG$(AR$,1,Y1-1)&"4"&SEG$(AR$,Y1
+1,11-Y1)
486 CALL PAUSE(150):: RETURN
487 ! **SCHULDEN**
488 DISPLAY AT(15,1):A$(23)
489 DISPLAY AT(17,1):P$(P)&" ZAHLT DM:";
S(P):: K(P)=K(P)-S(P):: S(P)=0
490 CALL PAUSE(150):: RETURN
491 ! **ZINSEN**
492 DISPLAY AT(15,1):A$(22)
493 DISPLAY AT(17,1):P$(P):"SCHULDEN...D
M:";S(P):"+ ZINSEN...DM:";INT(S(P)/100):
"= GESAMT...DM:";S(P)+INT(S(P)/100)
494 S(P)=S(P)+INT(S(P)/100)
495 CALL PAUSE(150):: RETURN
496 DATA FARBEN,CHEMIE,ELEKTRO,VERSORGUN
G,FE-STAHL,MASCH.-BAU,AUTOMOBIL,BAUWERTE
,KAUFHAUS,BANKEN,VERSICHER.
497 DATA "ANKAUF NICHT MOEGLICH! KEINE A
KTIEN VORRAETIG!","ANKAUFMENGE ZU KLEIN/
GROSS!","KAPITAL REICHT NICHT AUS!"
498 DATA "VERKAUF NICHT MOEGLICH.      KE
INE AKTIEN VORHANDEN!","VERKAUFMENGE ZU
KLEIN/GROSS!","WARNUNG! KURSWERT ZU NIED
RIGBEI:"
499 DATA "ACHTUNG - DIVIDENDE!          KU
RSWERT ZU HOCH BEI:","AUTOMATISCHE DURCH
FUEHRUNG!","HAGERA","AUSGABE JUNGER AKTI
EN BEI:","ZINSEN:"
500 DATA "SCHULDEN:","SPERRMINORITAET LI
EGT VOR!","BITTE BESTAETIGEN SIE J/N:"
501 ! **MUSIKDATA**
502 DATA 200,110,220,330
503 DATA 300,120,440,440
504 DATA 300,440,330,220
505 DATA 250,160,320,450
506 DATA 400,300,350,400
507 DATA 250,700,350,110
508 DATA 0,0,0,0
509 SUB CHCOL(BGR,BCOL)
510 CALL SCREEN(BGR)
511 FOR Y1=1 TO 8 :: CALL COLOR(Y1,BCOL,
BGR):: NEXT Y1
512 SUBEND
513 SUB PAUSE(BRUCH)
514 FOR Y1=1 TO BRUCH :: IF BRUCH=50 THE
N CALL SOUND(-10,Y1*20+110,0)
515 CALL KEY(0,KEY3,STA3):: IF STA3=1 TH
EN 517
516 NEXT Y1
517 SUBEND
```

BUDDY JOE

Es ist 14:00 Uhr und Buddy Joe will noch schnell 100 DM zur Bank bringen, um sie für längere Zeit anzulegen. Er hat einen langen Weg und weiß, daß er sich beeilen muß, weil die Bank um 17:00 Uhr schließt. Doch zu allem Unglück hat er auch noch seine Kontonummer vergessen. Unterwegs fällt ihm ab und zu eine Ziffer der 4-stelligen Kontonummer ein. Drücke eine Ziffer zwischen 1 und 6, wenn Dich der Computer dazu auffordert. Je näher Du an der richtigen Kontonummer gewählt hast, desto länger legt Buddy Joe sein Geld an, was ihm natürlich auch mehr Zinsen einbringt. Durch ein Loch in der Tasche verliert er ständig Geld, kann es aber im Kampf gegen die Uhr wieder einsammeln. Buddy Joe muß unterwegs auch aufpassen, daß er nicht gegen einen Baum läuft, da er sonst 10 Minuten verliert. Hinter irgendeinem Baum lauert Black Jack, ein kleiner Gangster, der es auf Buddys Geld abgesehen hat. Trifft er ihn, kommt es zu einem Kampf, dessen Ausgang noch ungewiß ist. Sollte Black Jack gewinnen, stiehlt er Buddy 50 DM. Erreicht Buddy die Bank noch rechtzeitig vor Geschäftsschluß, kann er sein Geld anlegen und der Kassierer sagt ihm, auf wieviele Jahre dies möglich ist. Nun muß Buddy warten, bis die Zeit verstrichen ist, bevor er sein Geld einschließlich Zinsen wieder abholen kann. Der Betrag, den er dann ausgezahlt bekommt, ist auch gleichzeitig Deine erreichte Punktzahl.

Und nun mach' Dich auf den Weg, es ist gleich 14:00 Uhr.

Frajo Fry

```
100 REM********************
110 REM*                  *
120 REM*    Buddy Joe     *
130 REM* Copyright by     *
140 REM*  Frajo Fry       *
150 REM*                  *
160 REM* Bearbeitet vom   *
170 REM*    Team des      *
180 REM* Aktuell Verlages *
190 REM*     Muenchen     *
200 REM*                  *
210 REM*Benoetigte Geraete*
220 REM* TI99/4A  Konsole *
230 REM*                  *
270 REM* Speicherbelegung *
280 REM*    9342 Bytes    *
290 REM********************
295 REM
300 CALL CLEAR
310 PRINT ".........BUDDY..J
OE": : : : :".......EIN SPIE
L FUER": : :".......FINANZEX
PERTEN": : :
320 PRINT "......UND GLUECKS
PILZE": : : : : :".........
..C.BY": : :".........FRAJO.
.FRY"
330 FOR I=0 TO 12 STEP 2
340 FOR J=110 TO 1210 STEP 1
10
350 CALL SOUND(-10,J,I,J+5,I
,J+10,I)
360 NEXT J
370 NEXT I
380 REM SPIELANLEITUNG
390 CALL CLEAR
400 PRINT "BUDDY JOE MACHT S
ICH UM": :"14.00 UHR AUF DEN
 WEG ZUR": :"BANK, UM 100 DM
 LANGFRISTIG": :
410 PRINT "ANZULEGEN.UNTERWE
GS VERLIERT": :"ER GELD, DAS
 ER ABER WIEDER": :"EINSAMME
LN KANN. AUSSERDEM": :
420 PRINT "HAT ER SEINE 5-ST
ELLIGE KON-": :"TONUMMER VER
GESSEN, OHNE DIE": :"ER DIE
BANK NICHT BETRETEN": :
430 PRINT "KANN. NACH UND NA
CH ERINNERT": :"ER SICH AN D
IE ZIFFERN.": :"FORTSETZUNG
AUF 1"
440 CALL KEY(0,K,S)
450 IF S=0 THEN 440
460 IF K=49 THEN 470 ELSE 44
0
470 CALL CLEAR
480 PRINT "DU MUSST DIESE ZI
FFERN (VON": :"1-6) ERRATEN.
 JE NAEHER DU": :"AN DIE RIC
HTIGE ZAHL KOMMST,": :
490 PRINT "DESTO HOEHER IST
DIE ZIFFER": :"DER KONTONUMM
ER. DIE SUMME": :"DER ZAHLEN
 ERGIBT DIE ANZAHL": :
500 PRINT "DER JAHRE,DIE BUD
DY JOE SEIN": :"GELD FESTLEG
T. ACHTUNG: DIE": :"BANK SCH
LIESST UM 17.00 UHR.": :
510 PRINT "BUDDY JOE WIRD MI
T DEN": :"PFEILTASTEN GESTEU
ERT.": :"DAS SPIEL BEGINNT A
UF 1"
520 CALL KEY(0,K,S)
530 IF S=0 THEN 520
540 IF K=49 THEN 560 ELSE 52
0
550 REM FARBEN & CHARACTERS
560 CALL CLEAR
570 CALL SCREEN(15)
580 CALL COLOR(16,10,15)
590 CALL COLOR(15,6,16)
600 CALL COLOR(14,4,16)
610 CALL COLOR(13,13,15)
620 CALL COLOR(12,7,15)
630 CALL COLOR(11,5,15)
640 CALL COLOR(10,2,12)
650 CALL COLOR(2,15,15)
660 CALL CHAR(152,"003F55AAD
5AAD5AA")
670 CALL CHAR(153,"30FCAA55A
B55AB55")
680 CALL CHAR(144,"FF8784849
4848787")
690 CALL CHAR(145,"FFFF21212
121FFFF")
700 CALL CHAR(154,"03030F1F3
F7FFFFF")
710 CALL CHAR(155,"0000F0F8F
CFEFFFF")
720 CALL CHAR(136,"FFFFC0C0C
0C0FFFF")
730 CALL CHAR(137,"FFC3C3C3C
BC3C3C3")
740 CALL CHAR(128,"00005A66B
DA55A3C")
750 CALL CHAR(120,"181818181
818183C")
760 CALL CHAR(112,"183C3C187
E183C66")
770 CALL CHAR(113,"0090D67F7
FD69")
780 CALL CHAR(104,"3C4289998
989423C")
790 CALL CHAR(96,"003C3C3C3C
3C3C")
800 CALL CHAR(97,"00007E7E7E
7E")
810 CALL CHAR(98,"183C3C187E
183C66")
820 CALL CHAR(99,"0000000018
180000")
```

```
830 REM SPIELFELDAUFBAU
840 CALL HCHAR(3,4,97,26)
850 CALL HCHAR(20,4,97,26)
860 CALL VCHAR(6,4,96,14)
870 CALL VCHAR(4,29,96,14)
880 CALL HCHAR(4,3,154)
890 CALL HCHAR(4,4,155)
900 CALL HCHAR(5,3,136)
910 CALL HCHAR(5,4,137)
920 CALL HCHAR(18,29,152)
930 CALL HCHAR(18,30,153)
940 CALL HCHAR(19,29,144)
950 CALL HCHAR(19,30,145)
960 B=5
970 FOR I=1 TO 6
980 RANDOMIZE
990 A=INT(22*RND)+6
1000 CALL HCHAR(B,A,128)
1010 CALL HCHAR(B+1,A,120)
1020 NEXT I
1030 B=B+3
1040 IF B=20 THEN 1050 ELSE
970
1050 DR$="KONTONUMMER.......
.ZINSSATZ 5%"
1060 GOSUB 3930
1070 CALL HCHAR(19,28,45)
1080 E=40
1090 F=10
1100 FOR I=1 TO F
1110 RANDOMIZE
1120 A=INT(16*RND)+4
1130 B=INT(24*RND)+5
1140 CALL GCHAR(A,B,U)
1150 IF U=32 THEN 1160 ELSE
1120
1160 CALL HCHAR(A,B,E)
1170 NEXT I
1180 F=F-2
1190 E=E+1
1200 IF E=45 THEN 1220 ELSE
1100
1210 REM START & STEUERUNG
1220 ZEIT=0
1230 GELD=100
1240 C=53
1250 D=0
1260 P=13
1270 CALL SOUND(200,440,1)
1280 X=5
1290 Y=5
1300 CALL HCHAR(X,Y,112)
1310 CALL KEY(0,K,S)
1320 IF S=0 THEN 1310
1330 IF K=69 THEN 1370
1340 IF K=88 THEN 1420
1350 IF K=83 THEN 1470
1360 IF K=68 THEN 1520 ELSE
1310
1370 GOSUB 3760
1380 X=X-1
1390 IF X>3 THEN 1410
1400 X=5
1410 GOTO 1570
1420 GOSUB 3760
1430 X=X+1
1440 IF X<20 THEN 1410
1450 X=18
1460 GOTO 1410
1470 GOSUB 3760
1480 Y=Y-1
1490 IF Y>4 THEN 1410
1500 Y=6
1510 GOTO 1410
1520 GOSUB 3760
1530 Y=Y+1
1540 IF Y<29 THEN 1410
1550 Y=27
1560 REM NEUES FELD PRUEFEN

1570 CALL GCHAR(X,Y,U)
1580 IF U>119 THEN 1710
1590 IF U=104 THEN 2000
1600 IF U=99 THEN 2530
1610 IF U=40 THEN 2040
1620 IF U=41 THEN 2340
1630 IF U=42 THEN 2520
1640 IF U=43 THEN 2570
1650 IF U=44 THEN 2650
1660 IF U=45 THEN 2980
1670 CALL HCHAR(X,Y,112)
1680 GOSUB 4120
1690 GOTO 1310
1700 REM  GEGEN BAUM LAUFEN

1710 AA=113
1720 IF K=88 THEN 1780
1730 IF K=83 THEN 1810
1740 IF K=68 THEN 1840
1750 CALL HCHAR(X+1,Y,AA)
1760 X=X+113-AA
1770 GOTO 1860
1780 CALL HCHAR(X-1,Y,AA)
1790 X=X-113+AA
1800 GOTO 1860
1810 CALL HCHAR(X,Y+1,AA)
1820 Y=Y+113-AA
1830 GOTO 1860
1840 CALL HCHAR(X,Y-1,AA)
1850 Y=Y-113+AA
1860 IF AA=113 THEN 1870 ELS
E 1680
1870 DR$="BUDDY JOE IST GEGE
N EINEN BAUMGELAUFEN UND BLE
IBT 10 MINUTENBEWUSSTLOS LIE
GEN"
1880 GOSUB 3860
1890 FOR I=1 TO 10
1900 CALL SOUND(50,220,1)
1910 GOSUB 4030
```

```
1920 NEXT I
1930 GOSUB 4070
1940 FOR I=1 TO 10
1950 GOSUB 4120
1960 NEXT I
1970 AA=112
1980 GOTO 1720
1990 REM  GELD AUFSAMMELN
2000 CALL HCHAR(X,Y,112)
2010 GELD=GELD+1
2020 GOTO 1680
2030 REM KONTONUMMER
2040 CALL HCHAR(X,Y,112)
2050 P=P+1
2060 IF P>18 THEN 1310
2070 DR$="BUDDY JOE ERINNERT
 SICH AN....EINE ZIFFER SEIN
ER KONTONUMMERDRUECKE EINE Z
AHL VON 1 BIS 6"
2080 GOSUB 3860
2090 CALL KEY(0,M,S)
2100 IF S=0 THEN 2090
2110 RANDOMIZE
2120 N=INT(6*RND)+49
2130 IF M=N THEN 2200
2140 IF ABS(M-N)=1 THEN 2220
2150 IF ABS(M-N)=2 THEN 2240
2160 IF ABS(M-N)=3 THEN 2260
2170 IF ABS(M-N)=4 THEN 2280
2180 BB=49
2190 GOTO 2290
2200 BB=54
2210 GOTO 2290
2220 BB=53
2230 GOTO 2290
2240 BB=52
2250 GOTO 2290
2260 BB=51
2270 GOTO 2290
2280 BB=50
2290 CALL HCHAR(1,P,BB)
2300 D=D+BB-48
2310 GOSUB 4070
2320 GOTO 1680
2330 REM ZINSSATZ
2340 CALL HCHAR(X,Y,112)
2350 IF C=56 THEN 2440
2360 IF C=50 THEN 2400
2370 RANDOMIZE
2380 A=INT(10*RND)+1
2390 IF A>5 THEN 2440
2400 DR$="DER ZINSSATZ STEIG
T UM 1%"
2410 C=C+1
2420 GOSUB 3860
2430 GOTO 2470
2440 DR$="DER ZINSSATZ FAELL
T UM 1%"
2450 C=C-1
2460 GOSUB 3860
2470 GOSUB 4030
2480 GOSUB 4070
2490 CALL HCHAR(1,30,C)
2500 GOTO 1680
2510 REM STEIN UEBERSPRINGEN
2520 CALL HCHAR(X,Y,99)
2530 IF K=69 THEN 1380
2540 IF K=88 THEN 1430
2550 IF K=83 THEN 1480 ELSE
1530
2560 REM 10 DM GEFUNDEN
2570 CALL HCHAR(X,Y,112)
2580 DR$="BUDDY JOE FINDET A
UF SEINEM...WEG EINEN 10 DM
SCHEIN"
2590 GOSUB 3860
2600 GOSUB 4030
2610 GELD=GELD+10
2620 GOSUB 4070
2630 GOTO 1680
2640 REM UEBERFALL
2650 CALL HCHAR(X,Y,112)
2660 DR$="BUDDY JOE WIRD VON
 BLACK JACK UEBERFALLEN UND
ES KOMMT ZU...EINEM KAMPF"
2670 GOSUB 3860
2680 RANDOMIZE
2690 BB=INT(6*RND)+2
2700 FOR I=1 TO BB
2710 CALL HCHAR(X,Y,98)
2720 GOSUB 4030
2730 CALL HCHAR(X,Y,112)
2740 GOSUB 4030
2750 NEXT I
2760 IF BB/2=INT(BB/2)THEN 2
910
2770 CALL HCHAR(X,Y,98)
2780 GOSUB 4070
2790 DR$="BLACK JACK HAT GEW
ONNEN UND...STIEHLT BUDDY JO
E 50 DM"
2800 GOSUB 3860
2810 GOSUB 4030
2820 GOSUB 4070
2830 CALL HCHAR(X,Y,112)
2840 GELD=GELD-50
2850 IF GELD>0 THEN 2870
2860 GELD=0
2870 FOR I=1 TO BB
2880 GOSUB 4120
2890 NEXT I
2900 GOTO 1680
2910 GOSUB 4070
2920 DR$="BUDDY JOE HAT GEWO
NNEN UND....BLACK JACK VERSC
HWINDET"
2930 GOSUB 3860
2940 GOSUB 4030
2950 GOSUB 4070
2960 GOTO 2870
2970 REM BANK ERREICHT
```

```
2980 CALL HCHAR(X,Y,112)
2990 IF P>17 THEN 3060
3000 CALL SOUND(200,110,1)
3010 CALL HCHAR(19,28,45)
3020 X=5
3030 Y=5
3040 CALL HCHAR(X,Y,112)
3050 GOTO 1680
3060 DR$="BUDDY JOE ERREICHT
 DIE BANK"
3070 GOSUB 3860
3080 CALL HCHAR(19,28,32)
3090 IF GELD<100 THEN 3120
3100 DR$="ER LEGT "&STR$(GEL
D)&" DM FUER "&STR$(D)&" JAH
RE..ZU "&STR$(C-48)&"% AN"
3110 GOTO 3130
3120 DR$="ER LEGT "&STR$(GEL
D)&" DM FUER "&STR$(D)&" JAH
RE...ZU "&STR$(C-48)&"% AN"
3130 GOSUB 3910
3140 GOSUB 4030
3150 GOSUB 4070
3160 REM ZWISCHENSPIEL
3170 CALL HCHAR(1,2,32,31)
3180 DR$="INZWISCHEN SAMMELT
 BLACK JACK DAS GELD EIN DAS
 BUDDY JOE....VERLOREN HAT"
3190 GOSUB 3860
3200 YY=29
3210 YY=YY-1
3220 FOR XX=4 TO 19
3230 CALL GCHAR(XX,YY,U)
3240 IF U>119 THEN 3270
3250 CALL HCHAR(XX,YY,98)
3260 CALL HCHAR(XX,YY,32)
3270 NEXT XX
3280 YY=YY-1
3290 FOR XX=19 TO 4 STEP -1
3300 CALL GCHAR(XX,YY,U)
3310 IF U>119 THEN 3340
3320 CALL HCHAR(XX,YY,98)
3330 CALL HCHAR(XX,YY,32)
3340 NEXT XX
3350 IF YY=5 THEN 3360 ELSE
3210
3360 GOSUB 4070
3370 REM NACHSPANN
3380 DR$="......."&STR$(D)&"
 JAHRE SPAETER"
3390 GOSUB 3930
3400 X=5
3410 Y=5
3420 CALL HCHAR(X,Y,112)
3430 DR$="BUDDY JOE GEHT WIE
DER ZUR BANK"
3440 GOSUB 3860
3450 FOR I=1 TO 14
3460 CALL HCHAR(X,Y,32)
3470 X=X+1
3480 CALL HCHAR(X,Y,112)
3490 FOR J=1 TO 40
3500 NEXT J
3510 NEXT I
3520 FOR I=1 TO 23
3530 CALL HCHAR(X,Y,32)
3540 Y=Y+1
3550 CALL HCHAR(X,Y,112)
3560 FOR J=1 TO 40
3570 NEXT J
3580 NEXT I
3590 FOR I=1 TO D
3600 GELD=GELD+GELD/100*(C-4
8)
3610 NEXT I
3620 GELD=INT(GELD*100)/100
3630 CALL HCHAR(19,28,32)
3640 GOSUB 4070
3650 CALL COLOR(2,2,15)
3660 DR$="BUDDY JOE HEBT SEI
N GELD AB...INSGESAMT "&STR$
(GELD)&" DM"
3670 GOSUB 3860
3680 GOSUB 4030
3690 GOSUB 4070
3700 DR$="BUDDY JOE GIBT FUE
R ALLE SEINEFREUNDE EIN GROS
SES FEST UND  LAEDT AUCH DIC
H HERZLICH EIN"
3710 GOSUB 3860
3720 GOSUB 4030
3730 GOSUB 4070
3740 GOTO 4340
3750 REM GELD VERLIEREN
3760 IF GELD=0 THEN 3830
3770 RANDOMIZE
3780 A=INT(100*RND)
3790 IF A>15 THEN 3830
3800 CALL HCHAR(X,Y,104)
3810 GELD=GELD-1
3820 RETURN
3830 CALL HCHAR(X,Y,32)
3840 RETURN
3850 REM DRUCKROUTINE
3860 CALL SOUND(300,440,1)
3870 CALL SOUND(300,660,1)
3880 CALL SOUND(300,550,1)
3890 V=22
3900 GOTO 3940
3910 V=23
3920 GOTO 3940
3930 V=1
3940 W=2
3950 FOR J=1 TO LEN(DR$)
3960 CODE=ASC(SEG$(DR$,J,1))
3970 CALL HCHAR(V,W,CODE)
3980 W=W+1
3990 IF W<32 THEN 4020
4000 V=V+1
```

```
4010 W=2
4020 NEXT J
4030 FOR J=1 TO 400
4040 NEXT J
4050 RETURN
4060 REM TEXT LOESCHEN
4070 CALL HCHAR(22,2,32,31)
4080 CALL HCHAR(23,2,32,31)
4090 CALL HCHAR(24,2,32,31)
4100 RETURN
4110 REM UHRZEIT
4120 ZEIT=ZEIT+1
4130 IF ZEIT=60 THEN 4210
4140 IF ZEIT=120 THEN 4230
4150 IF ZEIT=150 THEN 4250
4160 IF ZEIT=165 THEN 4270
4170 IF ZEIT=180 THEN 4180 E
LSE 4320
4180 DR$="ES IST 17 UHR UND
DIE BANK....SCHLIESST FUER H
EUTE"
4190 CALL COLOR(15,6,2)
4200 GOTO 4280
4210 DR$="NOCH 2 STUNDEN BIS
 DIE BANK...SCHLIESST"
4220 GOTO 4280
4230 DR$="NOCH 1 STUNDE BIS
DIE BANK....SCHLIESST"
4240 GOTO 4280
4250 DR$="NOCH 30 MINUTEN BI
S DIE BANK..SCHLIESST"
4260 GOTO 4280
4270 DR$="NOCH 15 MINUTEN BI
S DIE BANK..SCHLIESST"
4280 GOSUB 3860
4290 GOSUB 4030
4300 GOSUB 4070
4310 IF ZEIT=180 THEN 4340
4320 RETURN
4330 REM NEUES SPIEL
4340 DR$="DRUECKE 1 FUER EIN
 NEUES SPIEL"
4350 GOSUB 3860
4360 CALL KEY(0,K,S)
4370 IF S=0 THEN 4360
4380 IF K=49 THEN 560
4390 CALL GCHAR(19,28,U)
4400 IF U=45 THEN 4440
4410 DR$="BUDDY JOE FREUT SI
CH SEHR AUF DEINEN BESUCH"
4420 GOSUB 3910
4430 GOSUB 4030
4440 END
```

BACKGAMMON

ZIEL DES SPIELES:
Ziel des Spieles ist es, alle seine Steine aus dem Spielfeld zu würfeln.

REGELN:
Auf dem Spielfeld gibt es 24 Felder. Jeder Spieler besitzt 15 Steine. Das Spiel wird mit zwei Würfeln gespielt. Ein Stein bewegt sich von einem Feld zum anderen in eine vorgeschriebene Richtung. Diese Richtung ist für Schwarz vom Feld 24 zum Feld 1 und für Weiß vom Feld 1 zum Feld 24. Wenn ein Feld mit mehr als einem Stein belegt ist, heißt dieses Feld Point. Wenn ein Feld mit nur einem Stein belegt ist, heißt dieses Feld Blot.
Das Spiel ist in zwei Phasen geteilt. Die erste Phase hat das Ziel, alle Steine in ein sogenanntes Heimfeld zu bringen. Das schwarze Heimfeld befindet sich zwischen den Feldern 1–6, das Weiße zwischen den Feldern 19–24. Die zweite Spielphase ist das sogenannte bear off. In diesem Teil ist das Ziel, alle Steine als erster aus dem Heimfeld zu ziehen.
Zuerst die Regeln für die erste Phase. Sie beginnen das Spiel. Der Computer hat automatisch die Grundstellung aufgestellt. Der Würfel erscheint in der rechten oberen Ecke. Der Computer würfelt automatisch. Sie ziehen den Werten der Würfel entsprechend in ihre Zugrichtung. Wenn sie zum Beispiel eine 5 und eine 6 gewürfelt hätten, könnten sie vom Feld 24 auf das Feld 18 ziehen oder sie könnten vom Feld 13 auf das Feld 7 ziehen. Damit wäre die 6 verbraucht. Nun können sie mit der 5 von 13 auf 8, aber nicht von 24 auf 19, da auf 19 ein Point des Gegners ist, der nicht besetzt werden kann. Sie können auch von 8 auf 3 ziehen. Doch damit machen sie einen Blot und riskieren, daß sie vom Gegner geschlagen werden.
Sie können jeden einzelstehenden Stein, also ein Blot, schlagen. Der geschlagene Stein des Gegners muß wieder für Weiß in das schwarze Heimfeld (Feld 1–6) und für Schwarz in das weiße Heimfeld (24–19) gewürfelt werden. Ein Beispiel: Ein schwarzer Stein steht auf der Bar. Sie habe 6 und 3 gewürfelt.
Auf dem Feld 19 steht ein Point. Also können sie mit der 6 die Bar, das ist das Feld außerhalb des Spielfeldes, wo sie gerade stehen, nicht verlassen. Auf dem Feld 22 steht ein Blot. Sie können jetzt die Bar verlassen und den Stein auf 22 schlagen. Dieser muß jetzt seinerseits auf die Bar und im nächsten Zug vom Computer in ihr Heimfeld gewürfelt werden. Wenn allerdings auf dem Feld 22 auch ein Point gestanden hätte, hätten sie nichts tun können. Auch wenn irgendwo anders ein Zug für sie möglich wäre, dürfen sie ihn nicht machen. Wenn sie die Bar nicht verlassen können, müssen sie ihren Zug abgeben, d.h., daß sie auf die Space Taste drücken müssen, bis der Computer anzeigt, daß er denkt. Der Code für die Bar ist 00.
Nun kommen wir zur zweiten Phase des Spieles, das sogenannte bear off. Diese Phase tritt ein, wenn alle Steine im Heimfeld sind. Für diese Phase gelten besondere Regeln. Wenn sie zum Beispiel eine 5 und eine 4 gewürfelt haben und das Feld 5 und das Feld 3 sind ein Point. Nun können sie entweder einen Stein vom Feld 5 herauswürfeln (Code 00) oder sie können, wenn das Feld 6 besetzt ist, einen Stein von dort auf das Feld 1 legen. Nun zur 4. Auf dem Feld vier ist kein Stein. Sie haben jetzt die Wahl, ob sie einen normalen Zug machen, z.B. von 5 nach 1 oder ob sie

ein Feld näher gehen und einen Stein vom Feld 3 ins Aus ziehen. Dies ist ihnen überlassen.
Viel Spaß

Backgammon –
Ein Spiel mit viel Intelligenz, aber auch mit viel Glück! Sie brauchen nur den TI-99/4A+TI Extended Basic und einen kühlen Kopf!

Der Aufbau des Programmes ist nicht weiter schwierig. Durch das Menue können sie den Computer auf eine gewisse Spielweise „festnageln". Man kann eine einfache Logik wählen, mit der eine Blitzpartie kein Problem ist. Bei der aufwendigeren Logik müssen sie sich schon mehr Zeit nehmen. Die Zugzeit liegt zwischen 10 und 60 Sekunden. Die Logik ist einfach: Der Computer generiert, wenn er am Zuge ist, jeden möglichen Zug. Diese Züge werden in das Sub W geleitet, in dem ihnen Wertigkeiten zugeordnet werden. Diese Wertigkeiten werden miteinander verglichen, und der Zug, der den höchsten Wert hat, wird ausgeführt.
So ist es gewährleistet, daß der Computer immer den seiner Meinung nach besten Zug nimmt, der in dieser Stellung möglich ist.

ALLGEMEINE HINWEISE:

1. Da die Eingabe der Züge über eine Key-Schleife ausgeführt wird, besteht für einige das Problem, daß die Auto-Repeattasten zu schnell ansprechen. Dafür dient eine Delay Schleife in der Zeile 810, die von jedem Anwender nach seinem Maß dosiert werden kann.
2. Die Fabcode, die in den Zeilen 550 und 560 eingelesen werden, gelten für den Kontrast eines Schwarz-Weiß Fernsehers. Für einen Farb-Fernseher müssen diese Code umgeschrieben werden. (Siehe Seite 11).
3. Wenn ein Punkt in der Graphic mit mehr als sechs Steinen beladen ist erscheint ein doppelter Punkt.
4. Bei der Endphase des Spieles ist es ratsam, die Würfel in Reihenfolge auszuführen, da der Computer im Zweifelsfall nicht entscheiden kann, welcher Wrüfel genommen wurde und so nimmt er den Ersten.
5. Falls beim Menue vergessen wurde die Schwierigkeit einzustellen, nimmt der Computer im Zweifelsfall immer Schwierigkeit 2 an.
6. Bei Benutzung der Spielweiseneinstellung ist darauf zu achten, daß der Sprite deutlich zum Stehen kommt und die Zahl dementsprechend angezeigt wird. Wenn dies unterlassen wird, stellt der Computer automatisch den momentan angezeigten Wert an.
7. Aus speicherkritischen Aspekten ist es möglich, daß Zeilen mit Überlänge auftauchen. Diese Zeilen müssen wie folgt eingegeben werden: Zuerst wird die Zeile soweit eingegeben, wie es geht. Danach wird Enter gedrückt. Nun wird die Zeile neu editiert. Danach kann man die weiteren fehlenden Zeichen eingeben.
8. In der Spielanleitung fehlen sämtliche Kommas. Dies liegt daran, daß die Data-Zeilen ohne Anführungszeichen keine Kommas im Text zulassen. Data-Zeichen mit Anführungszeichen sind aus speicherkritischen Aspekten nicht möglich.

ANMERKUNGEN ZUR BEWERTUNGSROUTINE (Sub W):

Variablen:

X1–Y1 : In diesen Variab len ist ein Zug von X1 nach Y1 gespeichert.

X2–Y2 : In diesen Variablen ist der zweite Zug von X2 nach Y2 gespeichert.

N(,) : Das Array N ist auf N(24,1) dimensioniert (siehe Zeile 530). Die erste Ziffer steht für die Nummer des Feldes, die mit der Nummer in der Graphic übereinstimmen. Der Code der Bar ist in diesem Fall 0. Die zweite Ziffer steht für die Farbe des Steines. In dem Fall der Bewertung ist es 1, da die Steine des Computers interessieren, die weiß sind. Die schwarzen Steine sind dementsprechend unter 0 zu finden. N(4,1) zum Beispiel zeigt den Wert der Steine an, die weiß sind und zugleich auf dem Feld 4 stehen.

E : Die Variable E beinhaltet den Wert der Züge von X1 nach Y1 und von X2 nach Y2. Dieser Wert wird nachher (außerhalb dieses Subprogrammes) mit dem anderen Wert des vorhergehenden Zuges verglichen. Der Wert der Variable E wird in diesem Subprogramm initialisiert.

SEC : Dieser Wert läuft über das Menue in dieses Subprogramm ein. Sie steuert die Spielweise des Computers. SEC steht für Secrurity, und sobald dieser Wert erhöht wird, fällt der Wert der Variablen E wenn ein Stein allein steht und damit geschlagen werden kann.

BEA : BEA steht für Beat und wurde auch im Menue initialisiert. Um so höher der Wert der Variablen BEA ist, um so höher ist der Wert der Variablen E, wenn ein gegnerischer Stein geschlagen wird.

SUBPROGRAMMES:

Sub E : Ausführung des Spielerzuges/Erzeugung von Warntönen, die bei einem falschen Zug oder beim Drücken der Space-Taste zu hören sind.
Sub F : Aufruf der Logiksysteme nach dem gewählten Schwierigkeitsgrad.
Sub G : Feststellung, ob eine der Parteien gewonnen hat.
Sub H : Ermittlung der zwei Würfelwerte und graphische Darstellung dieser.
Sub I : Zuggenerator mit Halbzugtiefe 1 (entspricht Spielstärke 1) + Bewertungsroutine
Sub J : Kontrolle der Spielstadien (hat nur einen Einfluß auf die Computertaktik)
Sub K : Eingabealgorhythmus
Sub L : Zuggenerator mit Halbzugtiefe 2 (entspricht Spielstärke 2)
Sub N : Spielstratigie für Spielende
Sub O : Rechnungsroutine für Steinpositionen
Sub S : Bildschirm löschen/Gewinn- oder Verlustanzeige
Sub W : Bewertungsroutine für Spielstärke 2
Sub X : Ausführung eines Zuges auf graphisch und Computer interner Weise
Sub Z : Überprüfungsroutine eines Spielerzuges

Sven Pirk

```
100 ! **********************
110 ! *     BACKGAMMON       *
120 ! *   COPYRIGHT BY       *
130 ! *     SVEN  PIRK       *
140 ! *   BEARBEITET VOM     *
150 ! *      TEAM DES        *
160 ! *  AKTUELL VERLAGES    *
170 ! *      MUENCHEN        *
180 ! * BENOETIGTE GERAETE   *
190 ! * TI 99/4A   KONSOLE   *
200 ! *  EXTENDED - BASIC    *
210 ! *    DSK oder  CS      *
220 ! *                      *
230 ! *  SPEICHERBELEGUNG    *
240 ! *    11489  BYTES      *
250 ! *                      *
260 ! **********************
270 DATA 2,3,COPYRIGHT BY SV
EN PIRK,3,10,10.7.1984,5,9,+
++++++++++,6,9,+BACKGAMMON+
,7,9,++++++++++
280 DATA 17,2,SPIELWEISE:,18
,2,----------,20,1,AGGRESSIV
,20,21,DEFENSIV,21,1,<______
______[]___________>
290 DATA 15,2,SCHWIERIGKEIT
<1/2>:
300 DATA 8,3,(TI-99/4A+TI EX
TENDED BASIC),11,4,SPIELANLE
ITUNG: ENTER
310 DATA WILLKOMMEN BEI BACK
GAMMON,====================
=======,,STEUERUNG DER SPIEL
ANLEI-,TUNG,----------------
------------
320 DATA DISER TEXT KANN DUR
CH,DRUECKEN EINER BELIE-,BIG
EN TASTE GESTOPPT WERDEN,UND
 STARTET WIEDER SO-,BALD DIE
SE TASTE LOSGE-
330 DATA LASSEN WIRD.,DURCH
DRUECKEN DER TASTE,E KEHREN
SIE WIEDER ZUM,MENUE ZURUECK
.,,STEUERUNG DES MENUE
340 DATA ------------------
---------,DURCH DRUECKEN DER
 TASTEN,1 UND 2 LEGEN SIE DI
E,SCHWIERIGKEIT FEST.DIE,VAR
IABLE SPIELWEISE DES
350 DATA COMPUTERS HAT NUR B
EI,SPIELSTUFE 2 EINE,AUSWIRK
UNG.SIE WIRD,UEBER DIE TASTE
N S UND D,GESTEUERT.
360 DATA FIXIERT WIRD DIE SP
IELWEISE,DURCH DIE TASTE SPA
CE,MIT DER DANN GLEICHZEITIG
370 DATA DAS SPIEL GESTARTET
 WIRD.,,STEUERUNG DES SPIELE
S,------------------------
--,DIE EINGABE ERFOLGT IN DE
R
380 DATA LINKEN OBEREN ECKE.
DAS,ZEICHEN > ZEIGT AN,DASS
SIE JETZT IHRE EINGABE,MACHE
N KOENNEN.EINE EINGABE,BEINH
ALTET EINEN ZUG UND
390 DATA DIE REIHENFOLGE IN
DER SIE,DIE ZUEGE EINGEBEN I
ST,EGAL.EINSTELLIGE
400 DATA ZAHLEN WERDEN MIT E
INER,VORANGESTELLTEN NULL,EI
NGEGEBEN.,SIE SPIELEN DIE SC
HWARZEN,STEINE UND IHRE ZUGR
ICH-,TUNG IST VON 24 NACH 1.
410 DATA DIE BAR BESITZT DEN
 CODE 00,DER AUCH BEIM HERAU
S-,ZIEHEN VERWENDET WIRD.
420 DATA FOLGENDE FUNKTIONEN
 HABEN,DIE TASTEN:,ENTER-WIE
DERHOHLUNG DES VOR-,!!!!!!HE
RGEHENDEN ZUGES.,SPACE-WIRD
GEDRUECKT BEI
430 DATA !!!!!!NICHT AUSZUFU
EHRENDEN,!!!!!!ZUG.,K-ERMOEG
LICHT DIE KORREKTUR,!!BIS ZU
 DREI EINGEGEBENEN,!!ZEICHEN
.,E-ABBRUCH DES PROGRAMMES.
440 DATA BEI FALSCHER EINGAB
E,ERTOENT EIN SUMMTON DER,AU
CH BEI DRUECKEN DER SPACE,TA
STE ZU HOEHREN IST.,DAS ZEIC
HEN < ZEIGT AN
450 DATA DASS DER COMPUTER G
ERADE DA-,BEI IST EINEN ZUG
ZU ENT-,WICKELN.DIE ANALYSE
KANN,DURCH LAENGERES DRUECKE
N DER,TASTE 7 VORZEITIG ABGE
-
460 DATA BROCHEN WERDEN(NICH
T AN-,WENDEN WENN EIN STEIN
AUF,DER BAR STEHT).,,,VIEL S
PASS,,,!!!DRUECKEN SIE EINE
TASTE
470 DATA 0000001818,03030000
0000C0C0,0303001818000C0C0,C3
C3000000000C3C3,C3C300181800C
3C3,DBDB00000000DBDB
480 DATA 103010101038,780878
404078,780878080878,48484878
0808,784078080878,7840784848
78
490 DATA 784808080808,784878
784878,784878080878,47C54545
45E7,42C6424242E7,47C1474444
E7
500 DATA 47C1474141E7,45C547
4141E1,47C4474141E7,47C44745
45E7,47C5414141E1,47C5474545
E7
510 DATA 47C5474141E7,E725E5
8585E7,E226E28282E7,E721E784
```

```
84E7,E721E78181E7,E525E78181
E1
520 DATA 12,1,5,8,0,3,6,0,5,
1,1,2,13,0,5,17,1,3,19,1,5,2
4,0,2
530 DATA c,i,q,i,c,i,q,i,c,i
,q,i,c,i,c,i,c,i,q,i,c,i,q,i
,c,i,q,i,c,i,c
540 DATA c,i,i,i,i,i,i,i,i,i
,i,i,i,i,c,i,c,i,i,i,i,i,i,i
,i,i,i,i,i,i,c
550 DATA c,i,c,i,q,i,c,i,q,i
,c,i,q,i,c,i,c,i,c,i,q,i,c,i
,q,i,c,i,q,i,c
560 DATA 6,4,80,5,6,20,70,5,
6,12,70,3,6,30,75,2,17,4,90,
5,19,12,75,3,17,20,75,5,20,3
0,70,2
570 CALL CLEAR :: CALL SCREE
N(2):: CALL COLOR(1,16,16)::
 FOR I=2 TO 14 :: CALL COLOR
(I,2,16):: NEXT I :: CALL CL
EAR
580 FOR I=1 TO 13 :: READ X,
Y,A$ :: DISPLAY AT(X,Y):A$ :
: NEXT I :: CALL SPRITE(#1,4
3,2,170,125)
590 CALL KEY(0,K,S):: IF S=0
 THEN CALL MOTION(#1,0,0)::
GOTO 620 ELSE IF K=83 THEN C
ALL MOTION(#1,0,-10)ELSE IF
K=68 THEN CALL MOTION(#1,0,1
0)
600 IF K=32 THEN RESTORE 470
 :: GOTO 700 ELSE IF K=13 TH
EN 640 ELSE IF K=69 THEN CAL
L CLEAR :: END
610 IF K=49 THEN SW=1 ELSE I
F K=50 THEN SW=2 ELSE GOTO 5
90
620 DISPLAY AT(15,22):SW ::
CALL POSITION(#1,Y,X):: BEA=
125-X :: SEC=X-125 :: IF SEC
<0 THEN SEC=0 ELSE IF BEA<0
THEN BEA=0
630 DISPLAY AT(24,3)SIZE(4):
BEA :: DISPLAY AT(24,24)SIZE
(4):SEC :: GOTO 590
640 CALL CLEAR :: CALL DELSP
RITE(#1):: FOR I=2 TO 14 ::
CALL COLOR(I,16,2):: NEXT I
:: CALL COLOR(1,2,2)
650 FOR I=1 TO 82 :: READ A$
 :: PRINT :: PRINT :: FOR X=
1 TO LEN(A$):: DISPLAY AT(23
,X):SEG$(A$,X,1)
660 CALL KEY(0,K,S):: IF S<>
0 THEN 670 ELSE 680
670 IF K=69 THEN RESTORE ::
GOTO 570 ELSE CALL KEY(0,K,S
):: IF S<>0 THEN 670
680 NEXT X :: NEXT I
690 CALL KEY(0,K,S):: IF S=0
 THEN 690 ELSE RESTORE :: GO
TO 570
700 DIM N(24,1):: CALL DELSP
RITE(#1):: CALL CHARSET :: C
ALL CLEAR
710 CALL SCREEN(2):: A$="FFF
FFFFFFFFFFFFF" :: B$="00000
01818" :: C$="0000006666"
720 CALL COLOR(2,16,14,3,16,
2,4,16,2,5,2,13,6,8,16,7,8,1
3,8,2,16)
730 CALL COLOR(9,16,14,10,14
,14,11,13,14,12,16,2,13,16,2
,14,16,2)
740 CALL CHAR(70,B$,71,C$,75
,B$,76,C$,99,A$,113,A$,105,A
$,80,B$,81,C$,90,B$,91,C$)::
 FOR I=1 TO 6 :: READ A$ ::
CALL CHAR(I+39,A$):: NEXT I
750 FOR I=1 TO 24 :: READ A$
 :: CALL CHAR(119+I,A$):: NE
XT I :: FOR I=1 TO 8 :: READ
 X,Y,Z :: N(X,Y)=Z :: NEXT I
760 CALL HCHAR(5,2,99,31)::
FOR I=2 TO 32 :: READ A$ ::
CALL VCHAR(6,I,ASC(A$),6)::
NEXT I
770 FOR I=2 TO 32 :: READ A$
 :: CALL VCHAR(12,I,ASC(A$),
4):: NEXT I :: FOR I=2 TO 32
 :: READ A$ :: CALL VCHAR(16
,I,ASC(A$),6):: NEXT I
780 CALL HCHAR(22,2,99,31)::
 FOR I=1 TO 8 :: READ X,Y,Z,
W :: CALL VCHAR(X,Y,Z,W):: N
EXT I :: FOR I=30 TO 4 STEP
-2
790 IF I=18 OR I=16 THEN 800
 :: V1=V1+1 :: CALL VCHAR(3,
I,119+V1)
800 NEXT I :: FOR I=4 TO 30
STEP 2 :: IF I=18 OR I=16 TH
EN 810 :: V2=V2+1 :: CALL VC
HAR(24,I,131+V2)
810 NEXT I
820 CALL E(N(,)):: CALL G(N(
,)):: CALL F(N(,),BEA,SEC,SW
):: GOTO 820 :: END :: SUB E
(N(,)):: CALL H(W(),C):: A,B
=0 :: FOR I=1 TO C :: IF I=3
 THEN W(1),W(2)=W(3)
830 CALL G(N(,)):: CALL K(A,
B):: IF A<>0 OR B<>0 THEN 85
0
840 FOR J=1 TO 20 :: CALL SO
UND(-200,110,0):: NEXT J ::
GOTO 880
850 S,D=0 :: CALL Z(N(,),A,B
```

```
,D,W(),S):: IF S<>1 THEN 870
860 FOR J=1 TO 20 :: CALL SO
UND(-200,200,0):: NEXT J ::
GOTO 830
870 CALL X(N(,),A,B,D,0)
880 NEXT I :: SUBEND :: SUB
H(W(),C):: RANDOMIZE :: W(1)
=INT(RND*6)+1 :: W(2)=INT(RN
D*6)+1 :: IF W(1)=W(2)THEN W
(3),W(4)=W(1):: C=4 ELSE C=2
890 CALL VCHAR(1,20,39+W(1))
:: CALL VCHAR(1,25,39+W(2)):
: SUBEND
900 SUB G(N(,)):: FOR J=0 TO
 1 :: FOR I=0 TO 24 :: IF N(
I,J)<>0 THEN 920
910 NEXT I :: CALL S(J)
920 NEXT J :: SUBEND
930 SUB K(A,B)
940 DISPLAY AT(1,1)SIZE(11)B
EEP:">" :: FOR I1=1 TO 2 ::
FOR I2=1 TO 2
950 CALL KEY(0,K,S):: IF S=0
 THEN 950
960 IF K=13 AND(A<>0 OR B<>0
)THEN 1010 ELSE IF K=75 THEN
 GOTO 940 ELSE IF K=32 THEN
A,B=0 :: GOTO 1010 ELSE IF K
=69 THEN CALL CLEAR :: END
970 IF K<48 OR K>57 THEN 950
980 CALL VCHAR(1,(3+I2)+I1*3
,K,1):: X$(I2)=STR$(K-48)::
FOR DELAY=1 TO 1 :: NEXT DEL
AY :: NEXT I2
990 IF I1=1 THEN A=VAL(X$(1)
&X$(2))ELSE B=VAL(X$(1)&X$(2
))
1000 NEXT I1
1010 DISPLAY AT(1,1)SIZE(11)
BEEP:"<" :: SUBEND
1020 SUB F(N(,),BEA,SEC,SW):
: CALL H(W(),C):: FOR K=1 TO
 C STEP 2 :: IF SW=1 THEN CA
LL I(N(,),K,W()):: GOTO 1050
1030 S=0 :: CALL J(N(,),S,W(
),K):: IF S=1 THEN 1050 ELSE
 CALL L(N(,),W(),K,BEA,SEC)
1040 IF W(1)<>0 THEN CALL I(
N(,),K,W())
1050 NEXT K :: SUBEND :: SUB
 L(N(,),W(),K,BEA,SEC):: IF
W(K)=W(K+1)THEN X=2 ELSE X=1
1060 FOR T=X TO 2 :: IF T=1
THEN A=W(K):: B=W(K+1)ELSE A
=W(K+1):: B=W(K)
1070 FOR I=0 TO 24 :: IF N(I
,1)=0 THEN 1170 ELSE IF A+I>
24 THEN 1180 ELSE IF N(I+A,0
)>1 THEN 1170
1080 N(I,1)=N(I,1)-1 :: N(I+
A,1)=N(I+A,1)+1 :: FOR J=0 T
O 24 :: IF N(J,1)=0 THEN 115
0 ELSE IF J+B>24 THEN 1160 E
LSE IF N(J+B,0)>1 THEN 1150
1090 N(J,1)=N(J,1)-1 :: N(J+
B,1)=N(J+B,1)+1
1100 E,D=0 :: IF N(0,1)>0 TH
EN 1130 ELSE CALL W(N(,),E,S
EC,BEA,I,I+A,J,J+B)
1110 IF A1=0 AND B1=0 AND A2
=0 AND B2=0 THEN 1120 ELSE I
F ER>=E THEN 1130
1120 A1=I :: B1=I+A :: A2=J
:: B2=J+B :: ER=E :: DI=D
1130 CALL KEY(0,KEY,STATUS):
: IF KEY=55 THEN N(J,1)=N(J,
1)+1 :: N(J+B,1)=N(J+B,1)-1
:: N(I,1)=N(I,1)+1 :: N(I+A,
1)=N(I+A,1)-1 :: GOTO 1200
1140 N(J,1)=N(J,1)+1 :: N(J+
B,1)=N(J+B,1)-1
1150 NEXT J
1160 N(I,1)=N(I,1)+1 :: N(I+
A,1)=N(I+A,1)-1
1170 NEXT I
1180 NEXT T
1190 IF A1=0 AND B1=0 AND A2
=0 AND B2=0 THEN SUBEXIT
1200 IF N(B1,0)=1 THEN D1=1
1210 IF B1=B2 THEN 1220 ELSE
 IF N(B2,0)=1 THEN D2=1
1220 CALL X(N(,),A1,B1,D1,1)
:: CALL X(N(,),A2,B2,D2,1)::
 A1,B1,A2,B2,I,J,A,B,E,D,D1,
D2,W(1),W(2)=0 :: SUBEND
1230 SUB Z(N(,),A,B,D,W(),S)
1240 IF A<0 OR A>24 OR B<0 O
R B>24 THEN 1370 ELSE IF A=0
 THEN 1250 ELSE IF B=0 THEN
1290 ELSE 1340
1250 FOR I=1 TO 2 :: IF W(I)
=0 THEN 1260 ELSE IF W(I)+B=
25 THEN 1270
1260 NEXT I :: GOTO 1370
1270 IF N(0,0)=0 THEN 1370 E
LSE IF N(B,1)=1 THEN D=1 ELS
E D=0
1280 IF N(B,1)>1 THEN 1370 E
LSE W(I)=0 :: S=0 :: SUBEXIT
1290 IF N(0,0)>0 OR N(A,0)=0
 THEN 1370
1300 FOR T=24 TO 7 STEP -1 :
: IF N(T,0)<>0 THEN 1370
1310 NEXT T :: FOR I=1 TO 2
:: IF W(I)=0 THEN 1320 ELSE
IF A-W(I)>0 THEN 1320 ELSE 1
330
1320 NEXT I :: GOTO 1370
1330 IF N(A,0)=0 THEN 1370 E
LSE W(I)=0 :: S=0 :: SUBEXIT
1340 IF N(0,0)>0 OR N(A,0)=0
```

```
 OR N(B,1)>1 THEN 1370 ELSE
IF N(B,1)=1 THEN D=1 ELSE D=
0
1350 FOR I=1 TO 2 :: IF W(I)
=0 THEN 1360 ELSE IF A-W(I)=
B THEN W(I)=0 :: S=0 :: SUBE
XIT
1360 NEXT I
1370 S=1 :: SUBEND
1380 SUB O(N(,),C,X,E,R,Y)
1390 IF E=1 AND C=0 THEN X=N
(0,(NOT R)+2)+6 :: IF R=0 TH
EN Y=16 ELSE Y=18 ELSE IF E=
0 AND C=0 THEN X=N(0,R)+5 ::
 IF R=0 THEN Y=18 ELSE Y=16
ELSE GOTO 1410
1400 SUBEXIT
1410 IF E=1 THEN 1420 ELSE 1
440
1420 IF N(C,R)>5 THEN IF C>1
2 THEN X=21-(N(C,R)-6)ELSE X
=6+(N(C,R)-6)ELSE IF C>12 TH
EN X=21-N(C,R)ELSE X=6+N(C,R
)
1430 GOTO 1450
1440 IF N(C,R)>6 THEN IF C>1
2 THEN X=22-(N(C,R)-6)ELSE X
=5+(N(C,R)-6)ELSE IF C>12 TH
EN X=22-N(C,R)ELSE X=5+N(C,R
)
1450 IF C>18 THEN Y=2*C-18 E
LSE IF C>12 THEN Y=2*C-22 EL
SE IF C>6 THEN Y=28-2*C ELSE
 Y=32-2*C
1460 SUBEND
1470 SUB X(N(,),A,B,D,R)
1480 CALL O(N(,),A,X,O,R,Y)
1490 IF N(A,R)>6 AND N(A,R)<
13 THEN 1500 ELSE 1520
1500 IF R=0 THEN IF A/2=INT(
A/2)THEN Z=70 ELSE Z=90 ELSE
 IF A/2=INT(A/2)THEN Z=80 EL
SE Z=75
1510 GOTO 1530
1520 IF A=0 THEN Z=99 ELSE I
F A/2=INT(A/2)THEN Z=113 ELS
E Z=99
1530 CALL VCHAR(X,Y,Z):: N(A
,R)=N(A,R)-1 :: IF B=0 THEN
1590 ELSE CALL O(N(,),B,X,1,
R,Y)
1540 IF R=0 THEN IF B/2=INT(
B/2)THEN Z=71 ELSE Z=91 ELSE
 IF B/2=INT(B/2)THEN Z=81 EL
SE Z=76
1550 IF N(B,R)>=6 AND N(B,R)
<=11 THEN 1570
1560 Z=Z-1
1570 CALL VCHAR(X,Y,Z):: N(B
,R)=N(B,R)+1 :: IF D=0 THEN
1590 ELSE CALL O(N(,),O,X,1,
R,Y):: IF R=0 THEN Z=75 ELSE
 Z=90
1580 CALL VCHAR(X,Y,Z):: N(O
,(NOT R)+2)=N(O,(NOT R)+2)+1
 :: N(B,(NOT R)+2)=0
1590 IF R=1 THEN DISPLAY AT(
1,2)SIZE(8)BEEP:A;B
1600 SUBEND
1610 SUB W(N(,),E,SEC,BEA,X1
,Y1,X2,Y2)
1620 FOR I=1 TO 24 :: IF N(I
,1)=1 THEN E=E-((I+10)+SEC)
1630 NEXT I :: IF N(Y1,O)=1
THEN E=E+10+BEA
1640 IF N(Y2,O)=1 THEN E=E+1
O+BEA
1650 E=E+((15-X1)+(15-X2))
1660 SUBEND
1670 SUB J(N(,),RETURN2,W(),
K)
1680 FOR I=O TO 18 :: IF N(I
,1)<>O THEN 1700
1690 NEXT I :: CALL N(N(,),W
(),K,1):: RETURN2=1 :: SUBEX
IT
1700 FOR I=O TO 24 :: IF N(I
,1)<>O THEN 1720
1710 NEXT I
1720 FOR J=24 TO I STEP -1 :
: IF N(J,O)<>O THEN SUBEXIT
1730 NEXT J :: IF N(O,O)<>O
THEN SUBEXIT ELSE CALL N(N(,
),W(),K,O):: RETURN2=1
1740 SUBEND :: SUB N(N(,),W(
),K,S):: FOR J=O TO 1 :: CAL
L G(N(,))
1750 IF S=O THEN 1760 ELSE 1
800
1760 FOR I=1 TO 24 :: IF N(I
,1)<>O THEN 1780
1770 NEXT I
1780 B=W(K+J)+I :: IF B>24 T
HEN B=O
1790 CALL X(N(,),I,B,O,1)::
GOTO 1830
1800 FOR I=25-W(K+J)TO 24 ::
 IF N(I,1)<>O THEN 1820
1810 NEXT I :: GOTO 1760
1820 CALL X(N(,),I,O,O,1)
1830 NEXT J :: SUBEND
1840 SUB I(N(,),K,W()):: CAL
L J(N(,),RETURN2,W(),K):: IF
 RETURN2=1 THEN SUBEXIT
1850 F=-1 :: FOR I=1 TO 2 ::
 F=F+1 :: C=W(K+F)
1860 IF N(O,1)=O THEN 1900
1870 IF N(C,O)<2 THEN 1890
1880 IF F=1 THEN 1990 ELSE I
F N(W(K+1),O)<2 THEN F=-1 ::
 C=W(K+1)ELSE 1990
```

```
1890 IF N(C,0)=1 THEN D=1 ::
 A=0 :: B=C :: GOTO 1980 ELS
E D=0 :: A=0 :: B=C :: GOTO
1980
1900 FOR J1=1 TO 24 :: IF N(
J1,0)=1 THEN IF J1-C>0 THEN
IF N(J1-C,1)<>0 THEN A=J1-C
:: B=J1 :: D=1 :: GOTO 1980
1910 NEXT J1 :: FOR J1=1 TO
24 :: IF N(J1,1)=1 THEN 1920
 ELSE 1930
1920 IF J1-C>0 THEN IF N(J1-
C,1)>2 THEN A=J1-C :: B=J1 :
: D=0 :: GOTO 1980
1930 NEXT J1 :: FOR J1=1 TO
24 :: IF N(J1,1)>0 THEN 1940
 ELSE 1950
1940 IF J1-C>0 THEN IF N(J1-
C,1)=1 OR N(J1-C,1)>2 THEN A
=J1-C :: B=J1 :: D=0 :: GOTO
 1980
1950 NEXT J1 :: FOR J1=1 TO
24 :: IF N(J1,1)<>0 THEN IF
J1+C<25 THEN 1960 ELSE 1990
ELSE 1970
1960 IF N(C+J1,0)=0 THEN A=J
1 :: B=J1+C :: D=0 :: GOTO 1
980
1970 NEXT J1 :: IF A=0 AND B
=0 AND D=0 THEN 1990
1980 CALL X(N(,),A,B,D,1)
1990 A,B,D,J1=0 :: NEXT I ::
 SUBEND
2000 SUB S(J)
2010 IF J=1 THEN A$="SCHADE,
DU HAST VERLOREN" ELSE A$="G
RATULIERE,DU HAST GEWONNEN"
2020 CALL CLEAR :: CALL SCRE
EN(8):: CALL CHARSET :: DISP
LAY AT(10,2):A$ :: PRINT "NO
CH EIN SPIEL?"
2030 CALL KEY(0,K,S):: IF S=
0 THEN 2030 ELSE IF K=78 THE
N CALL CLEAR :: END
2040 RUN :: SUBEND
```

BÖRSE

Verk. TI 99/4A + Expansions-Box + Karte für Box + Extended Basic + Handbuch (deutsch) + Rec.Kabel (Orig. TI) + TI Bücher (Tips + Tricks u.a. DM 900,–, Walter Hufler, Am Geistpförtchen 2, 6 Ffm. 1, Tel. 069/287781

TI 99/4A + ExBasic + Mini-Ferns. + Cass. Rec. + 2 Joyst. (TI) + 7 Spielmodule + Progr. a. Kass. + 4 Bücher ü. TI 99/4A und mehr, zus. nur 900,–, Tel. 06431/53131

Super-Grafik-Paket mit 5 Demo (Ex-B-)Plattenzeichner (Ex-B-)Duell im All (TI-B), Moonlander (Ex-B), Drehen v. Sprites (Ex-B). Infos bei frank. Rückumschlag R. G. Watzling 10, 8250 Dorfen

Lohnsteuer sparen! LJA 1984 mit dem TI. Lst. und Est.-Tabellen und weitere Prg. Info gegen Rückporto Dieter Schiller, Rachelstr. 10a, 8400 Regensburg

99/4A kpl. m. Box + RS232 + V24 + 32K + Disk + X-Bas. + Dat Ver + Anal + Epson RX 80 m. Kabel + v. Zubeh. VB 2900,–, Tel. 06041/5285

Cassettenprogr. f. TI 99/4A, Spiele, Adventure, Nutzprogr. n. Wahl, Infoliste g. frank. Briefumschlag T. Brandau, Troppauerstr. 6, 8552 Höchstadt/A. Tel. 09193/1820

Suche TI Ext. Basic Modul bis 150,–. Tel. 0271/334867

TI-Module zu verk. Munchman 60,–, Hustle 30,–, Tombstone 35,–, Parsec 60,–, Moonsweeper 70,–. Tel. 02473/7320

Verk. tolle Masch.Prgr. für MM, XB u. E/A Mod. u.a. dt. Textverarb. Lsite geg. DM 0,80 in Briefm. von M. Hofmann, Bauernwaldstr. 130, 7000 Stgt. 1

Achtung, suche Programme für TI 99/4A. Liste senden an: Dirk Zimmer, An der alten Kantine 2, 3163 Sehnde 3.

TI 99/4A + Joyst. + Pgm's u. RecKabel u.a. 180,–, Thorsten Reich, Fritz Remystr. 11, 6050 Offenbach

Verkaufe: Softwareset (Ex), 8 Programme mit Cass. gegen 10 DM-Schein, 5 Spiele + 3 prakt. Progr. z.B. Mathe, die erste 10 plus Softwarebonus. Module Attack, CARRACE, TOMBSTONE-CITY je 300,– DM bei V. Brose, Hamburgerstr. 1, 4750 Unna-Kgb.

TI 99/4A Datenverw. + Analyse-Modul (neu – original verpackt) 50,– DM, Tel. 02331/402220

Suche für TI 99/4A Disk-Laufwerk, Tel. 089/4703941

Suche Music Maker-Modul, bis 35 DM, Stefan Lauterbach, Kyfitting 6, 3380 Goslar

TI-Rechner-Menügesteuert. Programm mit 14 Rechenarten. Incl. Cass. + Porto nur DM 10,– im Umschlag an: D. Karbach, Remscheider Str. 18, 5650 Soligen 1

TI 99/4A + versch. Spiel-Module + Joyst. 350,–. Tel. 06752/5735. M. Führer, 6570 Kirm

Verk. gegen Höchstgebot folg. Module: Statistik (engl.), Buchf., Speech Editor, Invaders, Munch Man, Video Games 2. Multipl, TI-Writer (jew. + Disk), Rechnungsst. (nur Disk), sowie alle 11 Adv. Spiele auf Cass. (Mod. erforderl.). Angebote bis 3 Wo. n. Erscheinungsdat. an M. Humbert, Saphirring 30, 3008 Garbsen 1, Versand per NN.

4-Farben Printer-Plotter PP-A4, Centronics-Schnittstelle, DIN A4 — 699,—
Slim Line Laufwerk 5,25", 500 kByte - DS/DD — 498,—
Einbausatz für 2 Laufwerke in original TI-Extension-Box m. Manual — 95,—
Disc-Steuerkarte, DD/DS bis zu 4 Laufwerken anschließbar — 635,—
NEUE PRODUKTE FÜR TI-99/4A
Extended Basic mit deutschem Handbuch — 239,50
Grafik Extended Basic mit deutschem Handbuch — 299,—
32 k-RAM-Erweiterung, seitlich ansteckbar — 299,—
32 K-RAM-Erweiterung mit Centronics-Schnittst. — 399,—

Preise inkl. Mwst., Info gratis. Lieferung gegen Nachnahme od. Vorauskasse

albs-Alltronic B. Schmidt · Postfach 1130 · 7136 Ötisheim · Tel. (07041) 2747 · Telex 7263738 albs

FACHVERSAND für DFÜ

DFÜ-SET : 675,– o.TI99. Getestet in Happy Computer, Heft 5/85 !
o.Abb.: Drucker mit Anschluß an unsere V.24 Schnittstelle.

gvm JOCHEN GERHARDT & BETTINA VAN MEGERN GbR
HÖHENSTR. 74b, 4 Düsseldorf 1, TEL.: 0211-776577, 9-19 UHR
oder 02107-4533, 10-18 UHR

HeGa-Soft
Software, Hardware und Literatur
Postfach 1201 8044 Lohhof

ID Data 2
Das universelle Dateiprogramm
für den TI
(benötigt Disc + 32 K)

Nur DM 119,--

ID Konto
Kontenverwaltung und
doppelte Buchhaltung
auf dem TI 99/4 A
(benötigt Disc + 32 K)

Nur DM 119,--

TI Profi Set
ID Data 2 + ID Konto
(Siehe Einzelbeschreibungen)

Zusammen DM 198,–

ANZEIGENSERVICE

TI-99!

GEISER-Zubehör: Bekannt für Qualität! Ab sofort sind die erfolgreichsten ID-Programme auch im Ausland erhältlich. Versand gegen Nachnahme (ab sFr 100,— frei Haus). Preise in sFr.

ID-DATA*: 98,—
Super-Datenverwaltungsprogramm: Editieren, sortieren, selektieren, drucken; kompatibel zu TI-Writer, Mailmerge. 500 Einträge (10 Zeilen) je Disk.

ID-KONTO*: 98,—
Endlich ein Profi-Programm für Kontenverwaltung und **doppelte** Buchhaltung. Bilanz, Erfolg, Journal, 99 Konten, 1000 Buchungen/Disk.

PROFI-SET*: 168,—
Besteht aus ID-DATA + ID-KONTO.
*32K, 1 Disk, X-BASIC (Drucker). Mit Assembler-Routinen, Größter Bedienungskomfort. Sie werden staunen!

TI-LOGO-DT: 49,—
240 Seiten, viele Bilder und Programme. Vergleichsregister deutsch-franz-englisch im Anhang. Sonderauflage!

GEISER

TI99-Zubehör, Fabrikstr. 9
CH-3360 **Herzogenbuchsee** (Schweiz)

Bezugsquellen BRD:
REIS, Bullay 06542/2715
Alltronic, Oetish. 07041/2747
RADIX, Hamburg 040/441695

DIE NÄCHSTE TI REVUE ERSCHEINT AM 26. JUNI

ACHTUNG! TI-99/4A Besitzer – ACHTUNG! TI-99/4A Besitzer

Peripherie		Zubehör		Module	
orig. TI-Peripherie Erw.-Box mit Disk-Steuerkarte u. Laufwerk	1.480,-	Graphic Tableau	248,-	Extended Basic (deutsch)	248,-
32 K-Byte RAM	425,-	Cartridge Expander (für 3 Module)	128,-	Mini Memory	290,-
RS 232 Karte	398,-	Akustikkoppler-Dataphon	298,-	Terminal Emulator II	85,-
Externe Erweiterungen		DIN A4-4 Farb.-Drucker/Plotter	898,-	Multiplan	320,-
32 K-Byte RAM	299,-	Slim Line Disk-Laufwerk (DS DD)	650,-	Editor Assembler	220,-
32 K-Byte RAM mit Centronics-Interface	399,-	Einbausatz für 2 Slim Line Laufwerke in orig. Box	96,-	Spiele von 30,- bis z. B. Moonmine	99,- 65,-

Alle Preise incl. MwSt. zuzügl. 5,- DM Versandkosten. Lieferung per Nachnahme oder Vorkasse, ab 200,- DM versandkostenfrei.
Fordern Sie unsere kostenlose Preisliste an.

PSR REIS GmbH Programm-Service
5584 Bullay
Bergstraße 80
Telefon 06542/2715

MACHEN SIE MIT!
VERDIENEN SIE GELD MIT IHREM COMPUTER!

Haben Sie einen TI 99/A? Können Sie schon programmieren? Schreiben Sie Ihre eigenen Programme – in Basic und/oder Assembler? Dann bietet TI REVUE Ihnen die Möglichkeit, mit diesem Hobby Geld zu verdienen.

Wie? Ganz einfach. Sie senden uns die Programme, die Sie für einen Abdruck als geeignet halten, zusammen mit einer Kurzbeschreibung, aus der auch die verwendete Hardware – eventuelle Erweiterungen – benutzte Peripherie – hervorgehen muß (Schauen Sie sich dazu den Kopf unserer Programmlistings an.)

Benötigt werden: Zwei Listings des Programms sowie eine Datenkassette oder Diskette! Wenn die Redaktion sich überzeugt hat, daß dieses Programm läuft und sich zum Abdruck eignet, zahlen wir Ihnen pro Programm je nach Umfang bis zu DM 300,–!

Sollten Sie keinen Drucker haben, genügt der Datenträger.

Sie erhalten Ihre Kassette selbstverständlich zurück, wenn Sie einen ausreichend frankierten Rückumschlag mit Ihrer Adresse beifügen.

Bei der Einsendung müssen Sie mit Ihrer Unterschrift garantieren, daß Sie der alleinige Inhaber der Urheber-Rechte sind! Benutzten Sie bitte anhängendes Formular! (Wir weisen darauf hin, daß auch die Redaktion amerikanische und englische Fachzeitschriften liest und „umgestaltete" Programme ziemlich schnell erkennt).

Um Ihnen die Arbeit zu erleichtern, finden Sie hier ein Formular. Sie können es ausschneiden oder fotokopieren.

PROGRAMMANGEBOT

Name des Einsenders: _____

Straße/Hausnr./Tel.: _____

Plz/Ort: _____

Hiermit biete ich Ihnen zum Abdruck folgende(s) Programm(e) an:

_____

_____

_____

_____

Benötigte Geräte: _____

_____

Beigefügt ( ) Listings ( ) Kassette ( ) Diskette

Ich versichere, der alleinige Urheber des Programmes zu sein!
Hiermit ermächtige ich die Redaktion, dieses Programm abzudrucken. Sollte es in den Kassetten-Service aufgenommen werden, erhalte ich auch dafür eine entsprechende Vergütung.

_____

Rechtsverbindliche Unterschrift

TI REVUE
Postfach 1107
8044 Lohhof

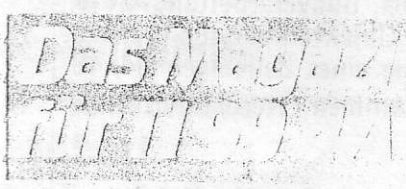
l + special + special + special

SONDERHEFT NR. 2/85
DM 14,80/ÖS 124/SFR 14,80

Anwender-Programme
Utilities
Schul-Programme
Spiele
Adventures

Rund 150 Seiten Listings für den 99/4A

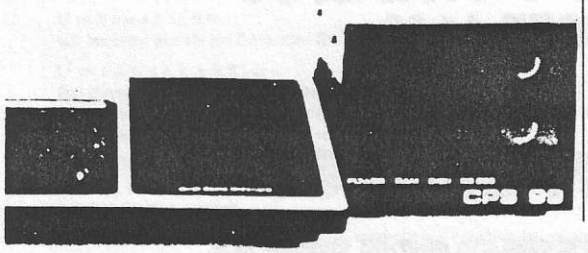
SPIEL-SALON
FÜR BUCHHALTER
MUSIKER &
SAMMLER
MATHE-TABELLEN
TORE
DIE HANDWERKER
KOMMEN
FÜR JEDEN ETWAS
WAS MAN SCHWARZ
AUF WEISS BESITZT
SPRITE-
KONSTRUKTEUR



Das Magazin für TI 99/4A

special + special + special + S

Neu für TI-99/4A



Peripherie

CPS 99 mit 1 Laufwerk DS DD (360 KB), Disk-Steuerkarte, 32 K-RAM, RS-232 und Centronics	1.698,-
wie vor mit 2 Laufw.	2.198,-
32 K-Byte RAM (extern)	279,-
32 K-Byte RAM (extern) mit Centronics-Interface	369,-
Graphic Tableau	248,-

Software

Mini Memory	290,-
Editor Assembler	198,-
Multiplan	298,-
TI-Writer	298,-
Basic Compiler	198,-
ID-Data	119,-
ID-Konto	119,-
Lagerverwaltung	98,-
Versandliste	98,-

Alle Preise incl. MwSt. zuzügl. 5,- DM Versandkosten. Lieferung per Nachnahme oder Vorkasse, ab 200,- DM versandkostenfrei.
Fordern Sie unsere kostenlose Preisliste an.

Programm-Service
5584 Bullay
Bergstraße 80
Telefon 06542/2715

Verlag und Redaktion haben eine neue Telefonnummer!
089/1298011 für Verlag
089/1298013 für Redaktion.

UNSER TELEFONSERVICE

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite
ACHTUNG!!!
Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33220 und für Listings/Programme: 089 / 129 80 13

Er steht für Forschung und Innovation an der Nahtstelle von Psychologie, Medizin und Informatik. Trotz finanzieller Engpässe konnten wir bei unseren Computern von vornherein nicht auf kompromißlose Zuverlässigkeit und höchste Flexibilität verzichten. So fiel die Wahl auf den TI99/4A. Sehen konnte der TI bisher noch nicht, doch dank unserem

Ist dieses Handicap nun beseitigt.

TI 99/4A mit angeschlossenem Video Digitizer AET-VD99. Verkleinerter Bildausschnitt. 64x48 Punkte, 16 Farb- bzw. Graustufen mit spezieller Software in druckbare 2-Farben-Darstellung umgewandelt

64000 Bildpunkte · 64 Grauwertstufen · belegt keinen Speicherplate INTI · 64 K eigenes RAM, auch als Erweiterung nutzbar · 32 K CMOS RAM (auf Wunsch zusätzlich integriert) · Incl. Bildverarbeitungs-Software · Darstellung der digitalisierten Bilder auf dem Bildschirm 256 x 192 Punkte · Objekterkennung (Lernen durch Zeigen) · Grauwerthistogramm · Kantendetektion · Ausschnittvergrößerung eines beliebigen 64x48 Punktebereiches in 16 Farbwerten · Schnittstellen zu verschiedenen Zeichenprogrammen · Hardcopyroutine für Epson FX80/RX80.

1498,- DM incl. MWSt.
ANDREAS EHLERDING TECHNOLOGIE
BERATUNG ENTWICKLUNG FERTIGUNG SERVICE
Kreisstraße 31 · D-3065 Nienstädt. · Tel. 05721/77 268

SONDERHEFT
Nr. 2/85

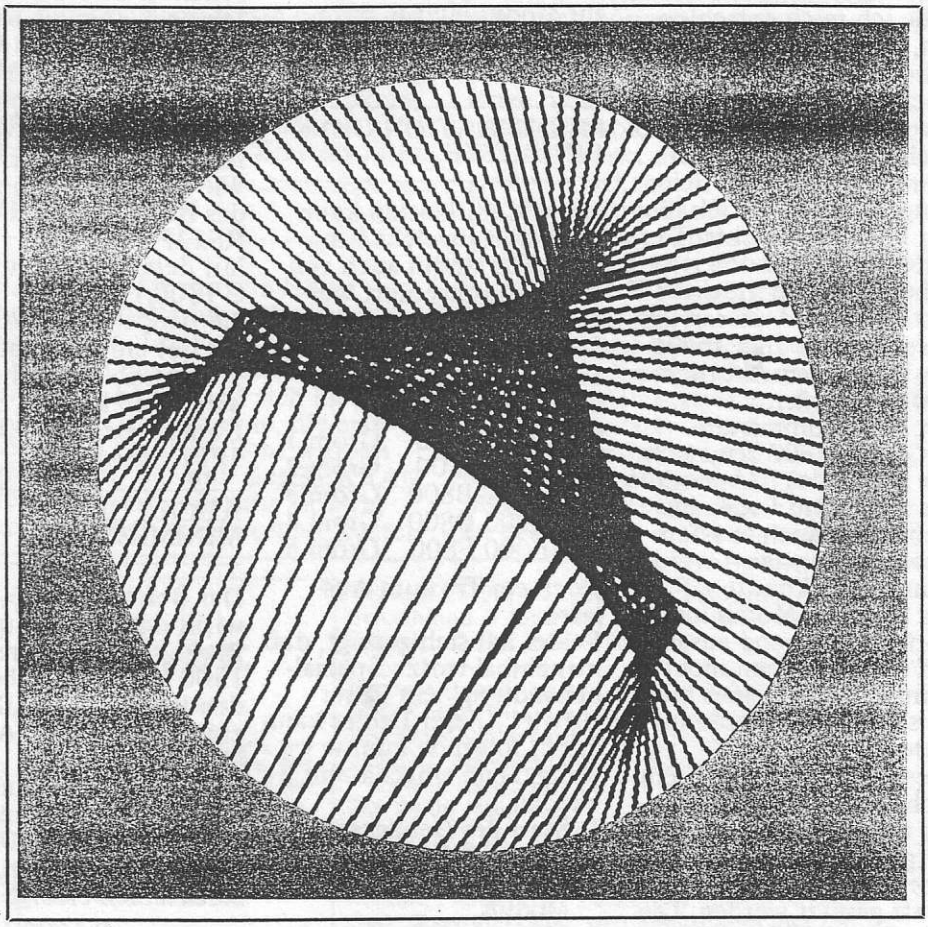
INHALT

UTILITIES SCHULE ANWENDER SPIELE IN TI BASIC

Panzer-Schlacht	12–17
Erdbeerpaul	18–20
Kompositionen	53–57
Apfelfänger	58–61
Seeschlacht	61–63
Space Invaders	69–72
Irvel	86–91
Copyright-Zeichen	91–93
Castle of Dome	94–96
Catch with Car	102–105
Skydriver	106–110
TI-Mühle	110–115
Dreieck- und Kreisberechnungen	115–117

UTILITIES SCHULE ANWENDER SPIELE IN EX BASIC

Disassembler	6–9
Reset	5
Komplexe Zahlen	6–8
Space 2000	9–11
3 x 3-Matrix	20–25
Softplot	25–29
Toreschießen	30–33
Tabellen-Berechnung	33–38
TI-Fall	39–43
Spickzettel	43–47
Advantage	48–49
Steuerberechnung	50–52
Prognose	64–66
Widerstände	67–69
Karnoug Veitch-Generator	77–81
Hexdump	82
Pokelisten-Generator	83
Ecarte	96–101
Boogaboo	188–123
Super-Grafikprogramm	124–127
Touch Typing Teacher	128–133
Sprite-Konstruktion-Set	134–139
Säulen-Diagramme	140–143



Impressum

TI-SPECIAL ist eine Sonderausgabe der TI-REVUE. Sie erscheint im TI/CBM-Verlag, Elisabethstraße 1, 8044 Lohhof.
Redaktion: Senator-Presse-Service.
Verantwortlich für den Inhalt: Hartmut König, Werner E. Seibt.
Verantwortlich für Anzeigen: Bruno Redase. Alle: Postfach 1107, 8044 Lohhof. Anfragen bitte nur schriftlich.
Vertrieb: Verlagsunion Wiesbaden.
Es gilt die Honorarliste des Verlages.
Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung. Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für einen einmaligen Abdruck und die Aufnahme in den Kassetten-Service. Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Jedwede Verwertung ist untersagt, Nachdruck nur mit ausdrücklicher schriftlicher Zustimmung des Verlages.

© by TI-CBM-Verlag, SPS und Autoren.

Erstverkaufstag dieser Ausgabe: 27. September 1985
Es gilt Preisliste Nr. 5 vom 1.1.1985.
Media-Unterlagen bitte anfordern bei Bruno Redase, co TI/CBM Verlag, Postfach 1107, 8044 Lohhof.
Für telefonische Anfragen: 089/129 80 11
Printed in Germany by Mayer & Söhne

TIPS & TRICKS

DER DEBUGGER UND SEINE MÖGLICHKEITEN

Aufgrund der Tatsache, daß die Zahl der Editor/Assembler-Besitzer laufend steigt und somit ebenfalls (hoffentlich) der Assembler Programmierer, habe ich mich entschlossen, Teile des Debuggers näher zu erläutern.
Nach genauem Studium, sowohl der englischen als auch der deutschen Beschreibung, zeigten sich einige Mängel, die ich hiermit beheben möchte. Der erste Befehl auf den ich näher eingehen möchte, gestattet das Auslesen des Assemblermoduls, sowie das Inspizieren sämtlicher Grom-Module.
Mit dem Kommando
M G6000,7800 zeigt sich der Inhalt des E/A-Moduls, das nach dem Laden des Debuggers auch aus dem Gromport entfernt werden kann. Jetzt könnte z.B. das Statistik-Modul ins Port gesteckt und ausgelesen werden, wenn nicht beim Einschub eines Moduls ein Resetimpuls auftreten würde.
Dieser Reset kann auf einfache Weise bei Modulen, die nur mit Groms (Graphic Read Only Memory) bestückt sind, außer Kraft gesetzt werden. Die nachfolgende Liste führt einige dieser Module auf:
Ti-Writer
Text- und Dateiverwaltung
Editor/Assembler
Datenverwaltung und Analyse
Statistic
Multiplan
Buchungsjournal
Wumpus
Speech-Editor
Amazing
Adventure usw.

Sollten Sie im Zweifel darüber sein, ob Ihr Modul ein reines Grom-Modul ist, genügt das Öffnen des Gehäuses.
Treten dabei nur ICs zutage mit maximal 16 Anschlüssen, so besteht kein Zweifel, daß es sich um ein nur mit Groms bestücktes Modul handelt. Zum Auslesen des Moduls genügt das Überkleben des Pins 1 mit einem Stück Tesafilm. Dieser Pin befindet sich, wenn Sie Ihr Modul von vorne betrachten (Steckerseite, die in den Gromschlitz gesteckt wird), rechts unten. Nach dieser Behandlung können Sie Ihr Modul jetzt in das Grom-Port stecken und auslesen.

MODIFIKATION OHNE EINGRIFFE

Diese Modifikation kann ohne irgendwelche Eingriffe ins Modul auch von außen durchgeführt werden.
Die nachfolgende Tabelle zeigt die gültigen Adreßbereiche der Groms:

M G6000,7800	Grom 1
M G8000,9800	Grom 2
M GA000,B800	Grom 3
M GC000,D800	Grom 4
M GE000,F800	Grom 5

Noch ein Tip zum Auslesen.
Leider hat sich gezeigt, daß manchmal trotz „Spezialbehandlung" ein Reset auftritt. Dieser Fehler kann durch folgende Punkte weitgehend eingedämmt werden.

1. Start des Debuggers
2. Befehl M 0,FFFF
3. Entfernen des E/A-Moduls
4. Einstecken des auszulesenden Moduls
5. Anhalten der Auflistung mit FCTN X
6. Auslesen laut obiger Tabelle
7. Bei Absturz Neu beginn mit Punkt 1

DSR AUSLESEN

Der zweite Befehl, auf den ich näher eingehen möchte, bezieht sich auf das Auslesen der Peripherie-Roms (DSR).
Die meisten Karten, u.a. P-Code, RS232, Controller, beinhalten ein Rom mit ihrem Betriebssystem. Diese Karten liegen auf folgenden CRU-Adressen.

1100	Controller
1300	RS232 I
1500	RS232 II
1800	Thermalprinter
1F00	P-Code

Der Adreßbereich des DSR-Roms befindet sich bei 4000 – 5FFF. Je nach Notwendigkeit wird vom System eines der Peripheriegeräte zu- oder abgeschaltet. Dieses Zuschalten bei Gebrauch bietet gegegenüber anderen Computersystemen den Vorteil, daß bei einer Peripherieerweiterung kein Rambereich verloren geht.
Wie kann man nun z.B. das Betriebssystem der RS232 I auslesen?
Mit dem Kommando
C 1300,1 <Enter> erfolgt die Ausgabe C 1300,XXXX die mit einer Eingabe von 1 <Enter> abgeschlossen wird. Nun können Sie das Rom im Bereich 4000–5FFF auslesen.

AUCH FÜR MINIMEMORY

Der Befehl C 1300,1 bewirkt dabei, daß nur ein Bit ab der Cru-Adresse 1300 Hex ausgegeben werden kann. C 1300,0 würde eine 16 Bit breite Ausgabe ab 1300 erlauben.
Bevor Sie nun ein anderes Rom auslesen, muß erst das Cru-Bit bei 1300 zurückgesetzt werden. Das Rücksetzen erfolgt mit:

	C 1300,1<Enter>
Ausgabe	C 1300,XXXX
Eingabe	0 <Enter>

Sollte jemand seine spielerische Ader mit dem Ein- bzw. Ausschalten der Peripherie-Leuchtdioden befriedigen wollen, so kann er dies durch das Senden eines Bits auf folgenden Adressen:

1100	/1100	Controller
130E	/1300	RS232 I
150E	/1500	RS232 II
1F00	/1F00	P-Code
– – –	/ – – – – – – – –	

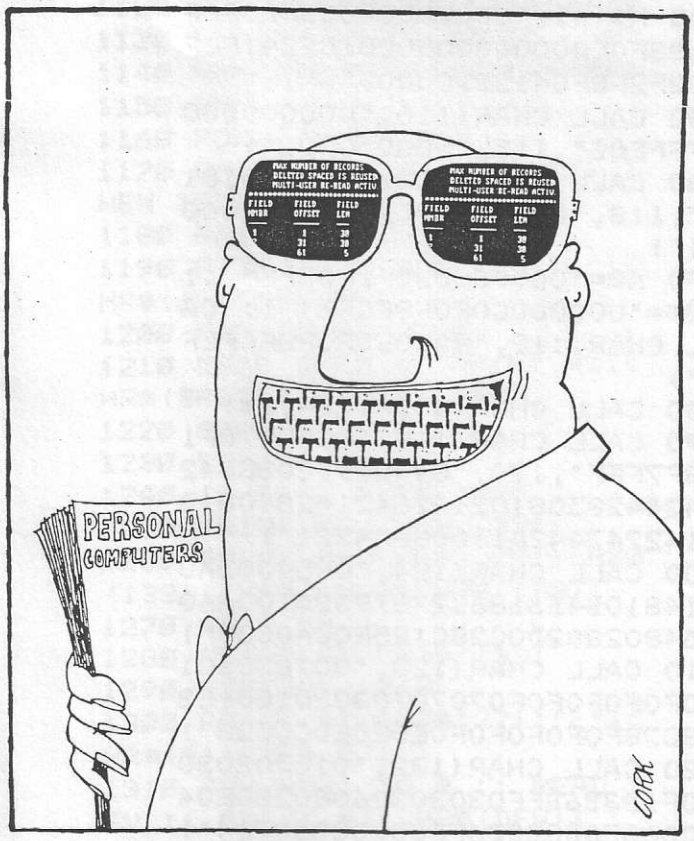
Led ein bzw. aus / Bit setzen zum Auslesen der DSR (DSR=Device Service Routine).
Aus dieser Tabelle wird erkennbar, daß sowohl bei der Controller- als auch der P-Code-Karte nur ein Bit für das Aufleuchten der Led und das gleichzeitige Auslesen gesetzt werden muß.
Zum Schluß möchte ich noch anmerken, daß auch Minimemory-Besitzer den Debugger starten und somit verwenden können.
Eine Kopie von einem Freund oder Bekannten wird sich bestimmt als nützlich erweisen.

Reiner Ziegler

ACHTUNG!!!

UNSER ☎ TELEFONSERVICE ☎

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite
ACHTUNG !!!
Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13



Space 2000

Der Wasserplanet ‚Atlantis' wird von einer Flotte feindlicher Raumschiffe angegriffen. 20 Gegner trachten nacheinander, den Planeten zu zerstören. Sie senken sich immer weiter hinab, bis sie die Verteidigungsbasis erreicht haben.

Sie, als joystickgewappneter Spieler, müssen versuchen, dies zu verhindern. Mit dem Fadenkreuz visieren Sie den Gegner an und zerstören ihn durch Betätigen des Feuerknopfes. Zu Beginn des Spieles haben Sie tausend Energieeinheiten zur Verfügung. Jeder Schuß kostet zehn Einheiten. Können Sie sich vor den Angreifern nicht mehr retten und bauen mit der SPACE-Taste einen Schutzschirm um die Basis auf, so müssen Sie dafür 100 Energieeinheiten opfern.

Die Invastion ist abgewehrt, wenn keines der 20 angreifenden Raumschiffe Ihre Basis zerstören onnte. Sollte alle 1000 Energieeinheiten vorzeitig verbraucht werden, ist das Spielvorzeitig beendet.

Jörg Bublath

Komplexe Zahlen

Das Programm bietet folgende mathematischen Operationen mit komplexen Zahlen: Transformation von Polar- in kartesische Koordinaten und umgekehrt, Addition, Subtraktion, Multiplikation, Division; Bilden von Wurzel, Kehrwert und Konjungiert kom. Zahl. Außerdem besteht die Möglichkeit, Zwischenergebnisse zu speichern.

Die Wahl der jeweilgen math. Operationen und die Art der Eingabe (z.B. in Kart.- oder Polarkoordinaten) erfolgt durch Steuern eines Pfeiles in einem Menüfeld.

Gerhard Hübner

```
100 ! ************************
110 ! *                      *
120 ! *     SPACE-2000       *
130 ! *                      *
140 ! *    Copyright by      *
150 ! *                      *
160 ! *   Joerg  Bublath     *
170 ! *                      *
190 ! * Benoetigte Geraete   *
200 ! *  TI99/4A  Konsole    *
210 ! *     Ext. Basic       *
220 ! *                      *
260 ! *  Speicherbelegung    *
270 ! *     8544 Bytes       *
280 ! *                      *
290 ! ************************
300 DIM M$(5)
310 CALL CLEAR :: CALL SCREE
N(2)
320 CALL CHARSET :: CALL DEL
SPRITE(ALL)
330 FOR I=2 TO 8 :: CALL COL
OR(I,16,1):: NEXT I
340 EN=1000
350 SC=0
360 ! TITEL BILD (J/N)
370 QW=QW+1 :: IF QW>1 THEN
770
380 !----------------------
390 !DEFINITIONEN FUER
400 !    TITELBILD
410 !----------------------
420 CALL CHAR(40,"FFFFFFFF")
430 CALL CHAR(41,RPT$("0F",8
),33,RPT$("F",16))
440 !----------------------
450 ! BILDSCHIRM AUFBAUEN
460 !----------------------
470 DATA 3,4,7,4,11,4,3,10,6
,10,3,15,6,15,3,20,11,20,3,2
5,7,25,11,25
480 FOR I=1 TO 12 :: READ A
:: READ B :: CALL HCHAR(A,B,
33,4):: NEXT I
490 DATA 15,4,19,4,22,4,15,1
1,22,11,15,18,22,18,15,25,22
,25
500 FOR I=1 TO 9 :: READ A :
: READ B :: CALL HCHAR(A,B,3
3,5):: NEXT I
510 DATA 4,10,4,15,4,18,4,20
,4,25,15,11,15,15,15,18,15,2
2,15,25,15,29
520 FOR I=1 TO 11 :: READ A
:: READ B :: CALL VCHAR(A,B,
33,8):: NEXT I
530 DATA 4,4,7,7,16,8,19,4,3
,13
540 FOR I=1 TO 5 :: READ A :
: READ B :: CALL VCHAR(A,B,3
3,4):: NEXT I
```

```
550 CALL CHAR(64,"0000000000
000010"):: DISPLAY AT(2,4):"
SCYPSOFT PRESENTS :" :: DISP
LAY AT(23,3):"COPYRIGHT BY J
@ BUBLATH"
560 DISPLAY AT(13,4):"PRESS
ANY KEY TO BEGIN"
570 !----------------------
580 ! BLINKEN DER BANDE
590 !----------------------
600 CALL COLOR(1,16,1)
610 CALL HCHAR(1,2,41,30)::
CALL HCHAR(24,2,41,30):: CAL
L VCHAR(1,2,40,24):: CALL VC
HAR(1,31,40,24)
620 CALL KEY(0,K,S):: IF S=1
 THEN 690 ELSE CALL COLOR(2,
7,2)
630 FOR I=1 TO 40 :: NEXT I
640 CALL KEY(0,K,S):: IF S=1
 THEN 690 ELSE CALL COLOR(2,
2,7)
650 FOR I=1 TO 40 :: NEXT I
660 GOTO 620
670 !    ABFRAGE FUER
680 !  SPIELANLEITUNG
690 CALL CLEAR :: DISPLAY AT
(7,1):"ANLEITUNG?" :: ACCEPT
 AT(7,12)VALIDATE("JN")SIZE(
1):AN$ :: IF AN$="J" THEN 17
60
700 DISPLAY AT(7,1):"INITIAL
ISIEREN-BITTE WARTEN"
710 !----------------------
720 ! DEFINITION DER
730 ! RAUMSCHIFFE UND
740 ! DES BILDSCHIRM-
750 ! AUFBAUES
760 !----------------------
770 CALL CHAR(96,"208A005402
48220000AA00228014401280240l
5004A00550208842005500A208")
780 CALL CHAR(100,"000000000
100010B0B0100010000000000000
00080008ODODO80008")
790 CALL CHAR(104,"183830706
0606060",105,"181C0C0E060606
06",106,"03030C0C3030C0C0")
800 CALL CHAR(107,"C0C030300
C0C0303",108,"0303060606060C
0C",109,"C0C0606060603030")
810 M$(1)="000040FF41033B6FF
F6F3B0341FF4000000003FE83C0C
0FDFEFDC0C083FE0300"
820 M$(2)="000000033F7FFF999
9FF7F3F0300000000000C0FCFEF
F9999FFFEFCC0000000"
830 M$(3)="00000000030F3FFFF
F3F0F0300000000040E1F3EFCF8F
0E0E0F0F8FC3E1F0E04"
840 M$(4)="00000000003F63FFF
F633F0000000000000001C2241FCF
8F2F2F8FC41221C0000"
850 CALL CHAR(110,"000000000
0FFFF00",115,"000010381")
860 CALL CHAR(112,"000000181
8",113,"00000018",114,"00000
01")
870 A$="000000030F1F3FFF" ::
 B$="000000C0F0F8FCFF" :: CA
LL CHAR(118,"80C0E0F0F8FCFEF
F")
880 CALL CHAR(33,A$,34,B$)
890 CALL CHAR(119,"0103070F1
F3F7FFF",120,"84422110088542
24244285081021428421428408l0
A142242442A11008844221")
900 CALL CHAR(124,"0054002A0
114810B4B3186327879330100AA0
054802882D0C28C1E5E0CA0C08")
910 CALL CHAR(128,"00787C3C1
E0F0F0F0F0F0707070301001E3
E3C78F0F0F0F0F0E0E0E0C0C08")
920 CALL CHAR(132,"010702030
60F1B3B6BFF030303060E0280E04
0C060F0D8DCD6FFC0C0C060704")
930 CALL CLEAR
940 CALL CHAR(116,RPT$("F",1
6),35,RPT$("F",16))
950 CALL CHAR(40,"80C0E0F0F8
FCFEFF",41,"0103070F1F3F7FFF
",42,"FF99E799E799E799")
960 FOR I=1 TO 40
970 !----------------------
980 !   BILDSCHIRMAUFBAU
990 !----------------------
1000 RANDOMIZE :: CALL HCHAR
((23*RND)+1,(31*RND)+1,(3*RN
D)+112):: NEXT I
1010 CALL HCHAR(24,1,116,32)
:: CALL HCHAR(23,7,116,18)::
 CALL HCHAR(22,7,116,18):: C
ALL HCHAR(21,7,116,18):: CAL
L HCHAR(20,7,116,18)
1020 CALL HCHAR(19,23,118)::
 CALL HCHAR(19,8,119):: CALL
 HCHAR(19,9,116,14):: CALL H
CHAR(18,11,119):: CALL HCHAR
(18,12,116,8)
1030 CALL HCHAR(18,20,118)::
 CALL COLOR(11,15,1):: CALL
HCHAR(23,1,35,6):: CALL HCHA
R(22,1,35,6):: CALL HCHAR(21
,1,35,6)
1040 CALL HCHAR(20,1,33,6)::
 CALL HCHAR(20,2,34):: CALL
HCHAR(20,4,34):: CALL HCHAR(
20,6,34):: CALL HCHAR(20,25,
33,8):: CALL HCHAR(20,26,34)
1050 CALL HCHAR(20,28,34)::
CALL HCHAR(20,30,34):: CALL
```

```
HCHAR(20,32,34):: CALL COLOR
(1,5,1)
1060 CALL HCHAR(21,25,35,8):
: CALL HCHAR(22,25,35,8):: C
ALL HCHAR(23,25,35,8)

1070 CALL HCHAR(17,17,40)::
CALL HCHAR(17,14,41):: CALL
HCHAR(16,15,41):: CALL HCHAR
(16,16,40):: CALL HCHAR(17,1
5,42,2):: CALL COLOR(2,9,1)
1080 EN=1000
1090 CALL HCHAR(17,12,104)::
 CALL HCHAR(17,19,105):: CAL
L HCHAR(16,12,108):: CALL HC
HAR(16,19,109):: CALL HCHAR(
15,13,106)
1100 CALL MAGNIFY(3)
1110 CALL HCHAR(15,18,107)::
 CALL HCHAR(14,14,110,4)
1120 DISPLAY AT(21,6)SIZE(15
):"ENERGIE:";EN
1130 DISPLAY AT(22,6)SIZE(15
):"SCORE  :";SC
1140 CALL SPRITE(#1,120,8,85
,114)
1150 CALL DELSPRITE(#2):: FO
R I=1 TO 90 :: NEXT I

1160 !----------------------
1170 ! ZUFAELLIGE BESTIMMUNG
1180 ! DER GESCHWINDIGKEIT
1190 ! UND DES AUSSEHENS
1200 ! DES FEINDLICHEN
1210 !   RAUMSCHIFFES
1220 !----------------------

1230 RANDOMIZE
1240 XXX=-(INT(30*RND)+6)
1250 SD=INT(4*RND+1)
1260 CALL CHAR(132,M$(SD))
1270 CALL SPRITE(#2,132,INT(
13*RND)+3,(80*RND)+1,1,2,XXX
)
1280 HIT=0
1290 CALL CHAR(33,A$,34,B$)
1300 GOSUB 1440
1310 CALL PATTERN(#1,120)::
CALL COLOR(#1,8)
1320 CALL KEY(0,C2,D2):: IF
C2=32 THEN CALL COLOR(10,11,
1):: SCH=1 :: EN=EN-100 ELSE
 CALL COLOR(10,1,1):: SCH=0
1330 CALL JOYST(1,C1,D1):: C
ALL MOTION(#1,-D1*3.5,C1*3.5
)
1340 CALL COINC(#2,115,114,2
0,HITS):: IF HITS THEN IF SC
H=0 THEN CALL DELSPRITE(#2):
: CALL SOUND(1000,-6,0):: CA
LL HCHAR(16,15,96,2)
1350 IF HITS=0 THEN 1400
1360 IF HITS AND SCH=1 THEN
1390
1370 IF HITS THEN CALL HCHAR
(17,14,98,4):: CALL COLOR(9,
9,1):: GOTO 1600
1380 GOTO 1400
1390 CALL PATTERN(#2,96):: C
ALL SOUND(500,-6,0):: FOR I=
1 TO 50 :: NEXT I :: GOTO 11
20

1400 CALL CHAR(34,A$,33,B$)
1410 GOSUB 1440
1420 DISPLAY AT(21,6)SIZE(15
):"ENERGIE:";EN :: IF EN<=0
THEN 1590
1430 GOTO 1290
1440 CALL JOYST(1,C1,D1)
1450 CALL POSITION(#2,A1,B1)
:: IF A1>125 THEN 1150
1460 CALL MOTION(#1,-D1*3.5,
C1*3.5)
1470 CALL KEY(1,C5,D5):: IF
C5=18 THEN 1520
1480 RETURN
1490 !----------------------
1500 !     SCHUSS !!!
1510 !----------------------
1520 CALL COLOR(2,9,1):: EN=
EN-10 :: CALL COINC(#1,#2,10
,HIT):: IF HIT THEN CALL PAT
TERN(#2,96):: CALL SOUND(300
,-5,0):: SC=SC+10 :: CALL MO
TION(#1,0,0)
1530 IF HIT=0 THEN CALL PATT
ERN(#1,100):: CALL COLOR(#1,
9)
1540 DISPLAY AT(22,6)SIZE(15
):"SCORE:";SC
1550 IF SC>190 THEN 1740
1560 IF HIT THEN 1150
1570 CALL JOYST(1,C1,D1):: C
ALL MOTION(#1,-D1*3.5,C1*3.5
)
1580 CALL COLOR(2,15,1):: RE
TURN
1590 HITS=-1 :: CALL SOUND(-
1000,-6,0):: CALL HCHAR(16,1
5,96,2):: CALL DELSPRITE(ALL
):: GOTO 1370
1600 CALL CLEAR
1610 !----------------------
1620 !     NIEDERLAGE  !!
1630 !----------------------
1640 DISPLAY AT(7,7):"DIE FE
INDLICHEN TRUPPEN  HA-BEN DI
CH BESIEGT": :"DEINE BASIS W
URDE ZERSTOERT": : : :"WILLS
T DU EINE ANDERE BASIS AUF A
TLANTIS VERTEIDIGEN ?"
1650 !----------------------
1660 !   NOCHMAL ABFRAGE
1670 !----------------------
```

```
1680 CALL DELSPRITE(ALL):: D
ISPLAY AT(17,1):"J=JA/N=NEIN
:" :: ACCEPT AT(17,14)SIZE(1
)VALIDATE("JNjn")BEEP:A$
1690 IF A$="J" OR A$="j" THE
N GOTO 310
1700 CALL CLEAR :: END
1710 !--------------------
1720 !       SIEG !!!
1730 !--------------------
1740 DISPLAY AT(7,7):"DU HAS
T UEBER DIE FEIND-.....LICHE
N ARMEEN GESIEGT!": :"BRAVO!
!!!!!!!!!!!!!!!!": : : :"MOEC
HTEST DU DIESEM PLANETENNOCH
 EINMAL HELFEN?"
1750 CALL CHAR(124,"10101010
10001010"):: CALL COLOR(12,1
6,1):: GOTO 1680
1760 CALL COLOR(2,16,1):: CA
LL CLEAR :: DISPLAY AT(1,8):
"ANLEITUNG:":TAB(8);"-------
---": :"DEINE AUFGABE IST ES
 EINE...BASIS AUF DEM WASSER
PLANETEN"
1770 DISPLAY AT(6,1):"ATLANT
IS ZU VERTEIDIGEN": :"LENKE
DEN CURSOR MIT DEINEM..JOYST
ICK AUF DAS ANGREIFENDERAUMS
CHIFF UND DRUECKE FEUER"
1780 DISPLAY AT(12,1):"SOLLT
EST DU DAS RAUMSCHIFF..NICHT
 TREFFEN SO KANNST DU..IM RI
CHTIGEN MOMENT FUER....EINIG
E SEKUNDEN EINEN"
1790 DISPLAY AT(16,1):"SCHUT
ZSCHIRM AKTIVIEREN,DER DICH
ALLERDINGS 100 ENERGIE-EINHE
ITEN KOSTET.EIN SCHUSS KOSTE
T NUR 10 EINHEITEN"
1800 DISPLAY AT(20,1):"ZWANZ
IG SCHIFFE GREIFEN.....INSGE
SAMMT AN,DANN HAST DU..SIE B
ESIEGT":"BITTE TASTE DRUECKE
N"
1810 CALL KEY(0,K,S):: IF S=
0 THEN 1810
1820 CALL CLEAR :: GOTO 700
```


Die nächste TI REVUE erscheint am 28. Oktober


```
100 ! **********************
110 ! *                    *
120 ! *  KOMPLEXE ZAHLEN   *
130 ! *                    *
140 ! *    Copyright by    *
150 ! *                    *
160 ! *  Gerhard Huebner   *
170 ! *                    *
180 ! *                    *
190 ! * Benoetigte Geraete *
200 ! *  TI99/4A  Konsole  *
210 ! *     Ext. Basic     *
220 ! *                    *
260 ! *  Speicherbelegung  *
270 ! *     5337 Bytes     *
280 ! *                    *
290 ! **********************
300 ON WARNING NEXT
310 CALL CLEAR :: CALL COLOR
(14,7,1):: CALL HCHAR(5,2,13
8,30):: CALL VCHAR(6,32,139)
:: CALL VCHAR(6,2,139,1):: C
ALL HCHAR(7,2,137,30)
320 CALL HCHAR(20,2,138,30):
: CALL VCHAR(21,2,139,3):: C
ALL VCHAR(21,32,139,3):: CAL
L HCHAR(24,2,137,30)
330 ZF=10 :: ZE=1
340 CALL CHAR(130,"C0F0FCFFF
FFCF0C",131,"0",132,"00183C7
E7E3C1800",91,"00002442FF422
4",80,"0102040810204080FF")
350 CALL CHAR(137,"FFFF",138
,"0000000000000FFFF",139,"C0C
0C0C0C0C0C0C")
360 DISPLAY AT(1,2):"1.einga
be von z=re+j(im)":" 2.einga
be von z=r P"
370 DISPLAY AT(3,2):"3.einga
be der anzeige":" 4.eingabe
aus s:"
380 DISPLAY AT(ZF,2):"1.z=z
(pol [ kar)" :: DISPLAY AT(Z
F+1,2):"2.z=z konjungiet kom
plex"
390 DISPLAY AT(ZF+2,2):"3.z=
1/z1" :: DISPLAY AT(ZF+3,2):
"4.z=wurzel(z1)" :: DISPLAY
AT(ZF+4,2):"5.z in den speic
her s"
400 DISPLAY AT(ZF+5,2):"6.z=
z1+z2" :: DISPLAY AT(ZF+6,2)
:"7.z=z1-z2" :: DISPLAY AT(Z
F+7,2):"8.z=z1*z2" :: DISPLA
Y AT(ZF+8,2):"9.z=z1/z2"
410 !
420 !**F U N K T I O. W A H
L
430 CALL HCHAR(ZF,3,130)
440 CALL KEY(0,K,S):: IF S=0
 THEN GOTO 440
```

```
450 CALL HCHAR(ZF,3,131)
460 IF K=69 THEN ZF=ZF-1 ::
IF ZF=9 THEN ZF=10
470 IF K=88 THEN ZF=ZF+1 ::
IF ZF=19 THEN ZF=18
480 IF K=13 THEN CALL SOUND(
10,2000,0):: CALL HCHAR(ZF,3
,132):: CALL GCHAR(ZF,4,FU):
: GOTO 510
490 GOTO 430
500 !
510 IF FU=49 OR FU=50 OR FU=
51 OR FU=52 OR FU=53 THEN EI
N=1 ELSE EIN=2
520 GOSUB 1120
530 IF FU=49 THEN GOSUB 690
540 IF FU=50 THEN GOSUB 750
550 IF FU=51 THEN GOSUB 820
560 IF FU=52 THEN GOSUB 890
570 IF FU=53 THEN GOSUB 640
580 IF FU=54 THEN GOSUB 970
590 IF FU=55 THEN A(I-1)=-A(
I-1):: B(I-1)=-B(I-1):: GOSU
B 970
600 IF FU=56 THEN GOSUB 1040
610 IF FU=57 THEN C(I-1)=1/C
(I-1):: D(I-1)=-D(I-1):: GOS
UB 1040
620 GOTO 430
630 !
640 !**           ** SPEICHER
650 AS=A(I-1):: BS=B(I-1)::
CS=C(I-1):: DS=D(I-1)
660 DISPLAY AT(6,1):"S:" ::
DISPLAY AT(6,3)SIZE(12):AS :
: DISPLAY AT(6,15):"+J" :: D
ISPLAY AT(6,17)SIZE(12):BS
670 RETURN
680 !
690 ! **      **POLAR<>KARTE
S
700 DISPLAY AT(23,1):""
710 DISPLAY AT(21,1):"Z=" ::
 DISPLAY AT(21,3):A(I-1):: D
ISPLAY AT(21,15):"+J" :: DIS
PLAY AT(21,17):B(I-1)
720 DISPLAY AT(22,1):"Z=" ::
 DISPLAY AT(22,3):C(I-1):: D
ISPLAY AT(22,16):"P" :: DISP
LAY AT(22,17):D(I-1)
730 RETURN
740 !
750 !**         ** VORZEICHEN
760 DISPLAY AT(23,1):"" :: D
ISPLAY AT(22,1):""
770 CALL VORZ(A(I-1),B(I-1),
C(I-1),D(I-1),E1,F1,G1,H1)
780 A(I-1)=E1 :: B(I-1)=F1 :
: C(I-1)=G1 :: D(I-1)=H1
790 DISPLAY AT(21,1):"Z=" ::
 DISPLAY AT(21,3):A(I-1):: D
ISPLAY AT(21,15):"+J" :: DIS
PLAY AT(21,17):B(I-1)
800 RETURN
810 !
820 !**       **    KEHRWERT
830 DISPLAY AT(23,1):"" :: D
ISPLAY AT(22,1):""
840 CALL REZI(A(I-1),B(I-1),
C(I-1),D(I-1),E1,F1,G1,H1)
850 A(I-1)=E1 :: B(I-1)=F1 :
: C(I-1)=G1 :: D(I-1)=H1
860 DISPLAY AT(21,1):"Z=" ::
 DISPLAY AT(21,3):A(I-1):: D
ISPLAY AT(21,15):"+J" :: DIS
PLAY AT(21,17):B(I-1)
870 RETURN
880 !
890 !**         **    WURZEL
900 DISPLAY AT(23,1):"" :: D
ISPLAY AT(22,1):""
910 !
920 CALL WURZ(A(I-1),B(I-1),
C(I-1),D(I-1),E1,F1,G1,H1)
930 A(I-1)=E1 :: B(I-1)=F1 :
: C(I-1)=G1 :: D(I-1)=H1
940 DISPLAY AT(21,1):"Z=" ::
 DISPLAY AT(21,3):A(I-1):: D
ISPLAY AT(21,15):"+J" :: DIS
PLAY AT(21,17):B(I-1)
950 RETURN
960 !
970 !**          **  ADDIERE
980 A(1)=A(1)+A(2):: B(1)=B(
1)+B(2)
990 CALL KARPOL(A(1),B(1),C(
1),D(1))
1000 DISPLAY AT(23,1):"" ::
DISPLAY AT(22,1):""
1010 DISPLAY AT(21,1):"Z=" :
: DISPLAY AT(21,3):A(1):: DI
SPLAY AT(21,15):"+J" :: DISP
LAY AT(21,17):B(1)
1020 RETURN
1030 !
1040 !**        **MULTIPLIZI
R
1050 C(1)=C(1)*C(2):: D(1)=D
(1)+D(2)
1060 CALL POLKAR(A(1),B(1),C
(1),D(1))
1070 DISPLAY AT(23,1):"" ::
DISPLAY AT(22,1):""
1080 DISPLAY AT(21,1):"Z=" :
: DISPLAY AT(21,3):A(1):: DI
SPLAY AT(21,15):"+J" :: DISP
LAY AT(21,17):B(1)
1090 RETURN
1100 !
1110 !****E I N G A B E ****
**
```

```
1120 FOR I=1 TO EIN
1130 CALL HCHAR(ZE,3,130)
1140 CALL KEY(0,K,S):: IF S=
0 THEN GOTO 1140
1150 CALL HCHAR(ZE,3,131)
1160 IF K=69 THEN ZE=ZE-1 ::
 IF ZE=0 THEN ZE=1
1170 IF K=88 THEN ZE=ZE+1 ::
 IF ZE=5 THEN ZE=4
1180 IF K=13 THEN CALL SOUND
(10,2000,0):: CALL GCHAR(ZE,
4,FE):: GOTO 1230
1190 GOTO 1130
1200 NEXT I
1210 RETURN
1220 !
1230 IF FE=49 THEN GOTO 1270
1240 IF FE=50 THEN GOTO 1300
1250 IF FE=51 THEN A(I+1)=A(
I):: B(I+1)=B(I):: C(I+1)=C(
I):: D(I+1)=D(I):: GOTO 1200
1260 IF FE=52 THEN A(I)=AS :
: B(I)=BS :: C(I)=CS :: D(I)
=DS :: GOTO 1200
1270 DISPLAY AT(23,1):"Z=...
.........+J............" ::
DISPLAY AT(22,1):""
1280 DISPLAY AT(21,1):"Z=" :
: ACCEPT AT(21,3)SIZE(12)BEE
P:A(I):: ACCEPT AT(21,17)BEE
P:B(I)
1290 CALL KARPOL(A(I),B(I),C
(I),D(I)):: GOTO 1200
1300 DISPLAY AT(23,1):"Z=...
.........P............" ::
DISPLAY AT(22,1):""
1310 DISPLAY AT(21,1):"Z=" :
: ACCEPT AT(21,3)SIZE(13)BEE
P:C(I):: ACCEPT AT(21,17)BEE
P:D(I)
1320 CALL POLKAR(A(I),B(I),C
(I),D(I)):: GOTO 1200
1330 !
1340 CALL POKA(A(I),B(I),C(I
),D(I),E(I),F(I),G(I),H(I))
1350 DISPLAY AT(19,1)SIZE(3)
:"Z=" :: DISPLAY AT(19,3):G(
I):: DISPLAY AT(19,16)SIZE(1
):"P" :: DISPLAY AT(19,17):H
(I)
1360 DISPLAY AT(20,1)SIZE(2)
:"Z=" :: DISPLAY AT(20,3):E(
I):: DISPLAY AT(20,15)SIZE(2
):"+J" :: DISPLAY AT(20,17):
F(I)
1370 !**********SUB POKA*****
**
1380 SUB POKA(A,B,C,D,E,F,G,
H)
1390 IF C=0 AND D=0 THEN CAL
L KARPOL(A,B,G,H):: E=A :: F
=B
1400 IF A=0 AND B=0 THEN CAL
L POLKAR(E,F,C,D):: G=C :: H
=D
1410 SUBEND
1420 !**********SUB POLKAR***
*
1430 SUB POLKAR(A,B,C,D)
1440 A=C*(COS(D*PI/180))
1450 B=C*(SIN(D*PI/180))
1460 SUBEND
1470 !***********SUB KARPOL*
*
1480 SUB KARPOL(A,B,C,D)
1490 C=SQR(A^2+B^2)
1500 IF A=0 AND B=0 THEN D=0
 :: GOTO 1560
1510 IF A=0 AND SGN(B)=1 THE
N D=90 :: GOTO 1560
1520 IF A=0 AND SGN(B)=-1 TH
EN D=-90 :: GOTO 1560
1530 IF B=0 AND SGN(A)=-1 TH
EN D=180 :: GOTO 1560
1540 D=ATN(B/A)*180/PI
1550 IF SGN(A)=-1 AND SGN(B)
=-1 OR(SGN(A)=-1 AND SGN(B)=
1)THEN D=D+180
1560 SUBEND
1570 !
1580 !***********SUB REZI***
1590 SUB REZI(A,B,C,D,E,F,G,
H)
1600 G=1/C :: H=-D
1610 CALL POLKAR(E,F,G,H)
1620 SUBEND
1630 !
1640 !***********SUB WURZ**
1650 SUB WURZ(A,B,C,D,E,F,G,
H)
1660 G=SQR(ABS(C))
1670 H=D/2
1680 CALL POLKAR(E,F,G,H)
1690 SUBEND
1700 !
1710 !**********SUB VORZ***
1720 SUB VORZ(A,B,C,D,E,F,G,
H)
1730 B=-B
1740 CALL KARPOL(A,B,G,H)
1750 E=A :: F=B
1760 SUBEND
```

ACHTUNG!
Neue Telefon-Nummer!
089/1298013 Ab 1.10.85!

RESET

Eine Basic-Pokeliste für das Assemblerprogramm „RESET". Benötigt werden das Ex-Basic Modul und die 32 K RAM Expansion. Die Routine wird im Direktmoduls mit CALL LINK („RESET") gestartet. Dieses Programm sucht und findet ein „gelöschtes" Basicprogramm in der 32 K RAM Expansion, setzt die notwendigen Zeige und startet dieses Programm automatisch. Damit ein ordnungsgemäßer Ablauf erfolgt, müssen einige Voraussetzungen erfüllt sein. Dies sind:

1. Das Basicprogramm muß über mindestens eine Zeile mit einer Zeilennummer größer als 255 verfügen. Dies dürfte aber in der Regel kein Problem sein.
2. Die zuersg gespeicherte (eingegebene) Zeile sollte eine Kommentarzeile sein, da sie eventuell verändert wird. Dies ist nicht immer identisch mit der kleinsten Zeilennummer. Zum Umstellen gibt es folgende Möglichkeiten:

a Für Diskettenbetrieb:
1) Eingabe einer Kommentarzeile mit kleinster Zeilennummer un und mindestens 20 (in Worten zwanzig) Zeichen.
2) Speichern des Programmes mit SAVE DSK1.PRG,MERGE
3) Eingabe von NEW
4) Eingabe von MERGE DSK1.PRG
5) Speichern des Programmes mit SAVE DSK1.PRG

b Alle anderen müssen einen umständlicheren Weg einschlagen:
1) Siehe a) 1)
2) Editieren des gesamten Programmes, d.h., nacheinander über die Cursortasten jede Zeile anwählen und dort mindestens ein Zeichen überschreiben. Die ENTER Taste zu drücken reicht nicht aus.

Es versteht sich ja fast von selbst, daß diese Kommentarzeile nicht mehr geändert werden darf. Beim Eingeben neuer Programme zuerst die Kommentarzeile und dann das restliche Programm eingeben.

Hier nun noch eine kurze Routine, die ein versehentliches Ändern erschwert. Im Direktmodus eingeben:

CALL PEEK (−32718, A,B)::CALL LOAD (A* 256+B−65539,0,0)

Zur großen Überraschung hat die erste Zeile nun die Nummer 0 und ist daher gegen Ändern geschützt. Durch Eingabe von „RES" erhält diese Zeile wieder einen normalen Wert.

Peter Vorwerk

Disassembler

Der Disassembler ist ein nützliches Werkzeug für Minimem-Benutzer, das aber auch für Extended Basic und/oder Memory Expansion geeignet ist. Man kann damit Maschinensprache-Routinen disassemblen und somit auch von Programmen, die mit dem Line-by-Line Assembler geschrieben sind, ein Listing erzeugen. Dabei werden die Symbole der symbolischen Adressen und die Namen der Minimem Utility Routinen eingesetzt.

Natürlich kann man auch beliebige Betriebssystem-Routinen des ROM übersetzen.

Außerdem können mit diesem Programm RAM/ROM-Bereiche in hexadezimaler Form angezeigt werden, ebenso Bereiche im VDP-RAM.

Als Zugabe gibt es die Möglichkeit, das RAM/ROM nach einem vierstelligen Hexwert zu durchsuchen.

Zum schnelleren Ablauf steht zum Umrechnen der Dezimalzahlen wahlweise eine Routine in Maschinensprache zur Verfügung, die jedoch 80 Bytes im Minimem belegt.

Als Hardcopy wird ein 80-stelliger Drucker über eine RS232-Schnittstelle benutzt.

Nach dem Starten erscheint ein Auswahl-Menü mit 6 Optionen:

1. Disassembler allgemein
2. Disassembler mit Symbolen und Utility-Namen
3. Auffinden eines Hexstrings
4. Dumpen aus RAM/ROM
5. Dumpen aus VDP-RAM
6. Programmende

Nachdem eine Option gewählt worden ist, erfolgen die Eingaben für Drucker und Maschinensprache-Routine (ja/nein). Danach werden die Anfangs- und Endadresse des gewünschten Bereiches eingegeben. Nach Erreichen der Endadresse wird gefragt, ob mit der gleichen Option weitergemacht oder das Programm beendet werden soll.

Bis auf Option 5 laufen alle Optionen auch unter Extended Basic, allrdings ist Option 2 wegen der fehlenden Tabellen nicht sinnvoll. Ebenso kann die Maschinensprachroutine natürlich nicht geladen werden, es sei denn, amn hat eine Memexp und ändert die Adressen. Das Format der Ausgabe ist aus den Mustern ersichtlich. Die Bildschirmausgabe ist ähnlich (zwei Zeilen pro Assembler-Befehl, 3 statt 8 2-Byte Blöcke beim Dump).

Erdbeer-Paul

Die Hauptrolle des Programms spielt Paul, der Zwerg. Seine Aufgabe besteht im Pflücken von 12 Erdbeeren. Diese kann er aufnehmen, indem er unter die Erdbeere geht und Taste Q gedrückt wird. Hat er eine Erdbeere aufgenommen, so wird dieses am unteren rechten Bildrand angezeigt. Die Erdbeere muß er dann zu dem Topf bringen. Steht er am am Topf, muß noch einmal die Taste S bedient werden, damit Paul die Erdbeere ablegt. Aber natürlich gibt es auch Gefahren: Eine Maus in der unteren Etage, Spinnen in der oberen und eine Biene. Stößt Paul mit einem dieser Tiere zusammen, ist das Spiel beendet (neuer Start: Drücken einer Taste). Paul. hab aber auch eine Chance, denn er besitzt 5 Schwerter, mit diesen kann er alle Tiere außer der Biene vertreiben. Pro vertriebenem Tier gibt es 15 Punkte, pro abgelegter Erdbeere 30 Punkte und 6 Zusatzschwerte. Sinkt die Zahl der Schwerte auf 0, dann ist das Spiel beendet. Sind alle Erdbeeren geerntet, beginnt eine neue Runde. Das Benutzen der Schwerter erfolgt über Taste , :links und Taste. :rechts.

Gunther Krämling

```
BF  EQU >7F60
BG  EQU >7F66
BY  EQU >7F94
GN  EQU >7FB4
SC  EQU >7FC6
NM  EQU >7FD6

7F66  C28B       BG MOV  R11,R10
7F68  04C0          CLR  R0
7F6A  0201 0001     LI   R1,1
7F6E  0205 7F60     LI   R5,BF
7F72  0206 0400     LI   R6,>0400
7F76  DD46          MOVB R6,*R5+
7F78  06A0 7FB4     BL   @GN
7F7C  C204          MOV  R4,R8
7F7E  0581          INC  R1
7F80  06A0 7FB4     BL   @GN
7F84  0288 0001     CI   R8,1
7F88  1605          JNE  BY
7F8A  06A0 7FC6     BL   @SC
7F8E  0A44          SLA  R4,4
7F90  06A0 7FC6     BL   @SC
7F94  0A84       BY SLA  R4,8
7F96  06A0 7FC6     BL   @SC
7F9A  0AC4          SLA  R4,12
7F9C  06A0 7FC6     BL   @SC
7FA0  0581          INC  R1
7FA2  0288 0001     CI   R8,1
7FA6  131D          JEQ  >7FE2
7FA8  0281 0004     CI   R1,4
7FAC  131A          JEQ  >7FE2
7FAE  06A0 7FB4     BL   @GN
7FB2  10F0          JMP  BY
7FB4  0420 6044  GN BLWP @NUMREF
7FB8  0420 601C     BLWP @XMLLNK
7FBC  1200          JLE  >7FBE
7FBE  C120 834A     MOV  @>834A,R4
7FC2  C1C4          MOV  R4,R7
7FC4  045B          B    *R11
7FC6  0884       SC SRA  R4,8
7FC8  D100          MOVB R0,R4
7FCA  0844          SRA  R4,4
7FCC  0284 000A     CI   R4,10
7FD0  1102          JLT  NM
7FD2  0224 0007     AI   R4,7
7FD6  0224 0030  NM AI   R4,48
7FDA  0A84          SLA  R4,8
7FDC  DD44          MOVB R4,*R5+
7FDE  C107          MOV  R7,R4
7FE0  045B          B    *R11
7FE2  0202 7F60     LI   R2,BF
7FE6  0420 6048     BLWP @STRASG
7FEA  045A          B    *R10
```


ACHTUNG!
Neue Telefon-Nummer!
089/1298013 Ab 1.10.85!


```
10 REM*********************
11 REM*                   *
12 REM*   DISASSEMBLER    *
13 REM*   Copyright by    *
14 REM*                   *
15 REM* Fritz-H. Mueller  *
16 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole  *
19 REM*  Minimem-Modul    *
20 REM*optional: Xbasic   *
21 REM*  32-K erweiterung*
22 REM*  Drucker          *
23 REM* Speicherbelegung  *
24 REM*    9440 Bytes     *
25 REM*********************
26 REM
100 CALL CLEAR
110 CALL SCREEN(15)
120 PRINT : : : : : : : :"
     DISASSEMBLER": : :
130 PRINT "    (C) F.H.MUELLE
R 1984 ": : : : : :
140 REM  AM BOERNEKEN 1
150 REM  3181 RUEHEN
160 REM  TEL.05367/886
170 DIM OP$(78),UNAM$(28),UA
D(28),SY$(40),SYA(40)
180 DEF TX(X)=INT(X-((X)=10)
*7)+48)
190 DEF RST(Y)=Y-INT(Y)
200 DEF UH(X)=INT(X/16)
210 DEF LH(X)=X-(UH(X)*16)
220 DEF UB2=INT(B(2)/64)
230 DEF LB2=B(2)-(UB2*64)
240 DEF TS=INT(LB2/16)
250 DEF SAD=B(1)*256+B(2)
260 DATA >," ","  ","","",(,),
R,*,-,+,,@
270 READ ZX$,L1$,L2$,ZK$,KA$
,KZ$,R$,ST$,M$,PL$,LL$,KLA$
280 DATA COC,CZC,XOR,XOP,LDC
R,STCR,MPY,DIV
290 DATA LIMI,IDLE,RSET,RTWP
,CKON,CKOF,LREX
300 DATA JMP,JLT,JLE,JEQ,JHE
,JGT,JNE,JNC,JOC,JNO,JL,JH,J
OP,SBO,SBZ,TB
310 DATA LI,,AI,,ANDI,,ORI,,
CI,,STWP,,STST,,LWPI
320 DATA BLWP,NEG,DEC,SETO,B
,INV,DECT,ABS,X,INC,BL,,CLR,
INCT,SWPB,
330 DATA SRA,SRL,SLA,SRC
340 DATA SZC,SZCB,S,SB,C,CB,
A,AB,MOV,MOVB,SOC,SOCB
350 FOR I=1 TO 78
360 READ OP$(I)
370 NEXT I
380 DATA 1,3,0,8,11,16,0,0,"
 "
```

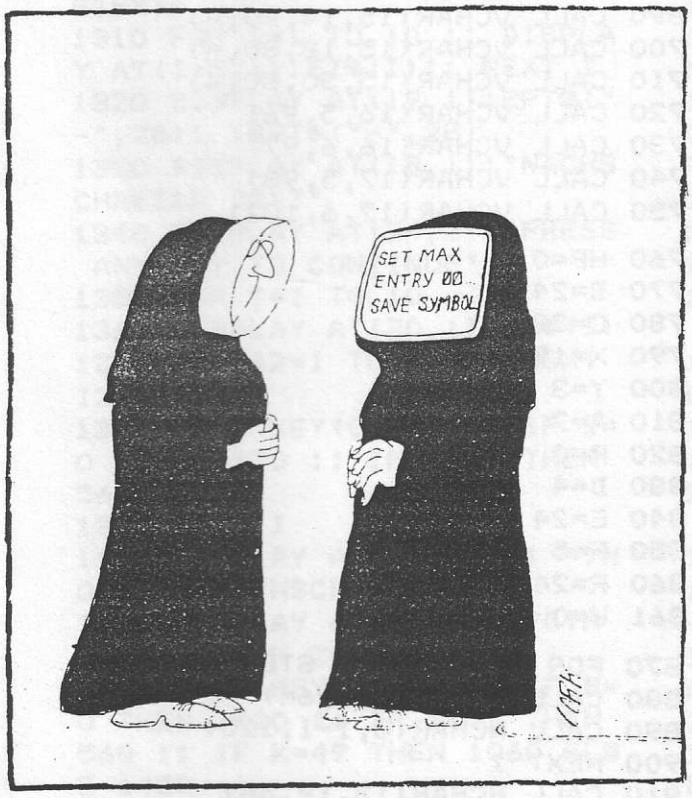
```
390 DATA 1,8,0,20,25,17,"  "
400 GOTO 1500
410 REM ***** COVERT BYTE TO
 HEX *****
420 ON AN3 GOTO 430,450
430 CALL LINK("TOX$",2,B(1),
B(2),BX$)
440 RETURN
450 FOR I=1 TO 2
460 A$=CHR$(TX(B(I)/16))
470 H$=CHR$(TX(RST(B(I)/16)*
16))
480 AH$(I)=A$&H$
490 NEXT I
500 BX$=AH$(1)&AH$(2)
510 RETURN
520 REM ***** CONVERT ADDRES
S TO HEX *****
530 ON AN3 GOTO 540,560
540 CALL LINK("TOX$",1,ADI,B
X$)
550 RETURN
560 B(1)=ADI/256
570 B(2)=256*(RST(B(1)))
580 GOTO 450
590 REM
600 REM ***** CONVERT TO DEC
IMAL *****
610 REM
620 ADO=0
630 FOR I=1 TO 4
640 AAS(I)=ASC(SEG$(AXI$,I,1
))
650 IF AAS(I)>57 THEN 680
660 AD(I)=VAL(CHR$(AAS(I)))
670 GOTO 690
680 AD(I)=AAS(I)-55
690 ADO=ADO+(AD(I)*16^(4-I))
700 NEXT I
710 RETURN
720 REM ***** INCR ADDRESS A
ND PEEK 2 BYTES *****
730 ADR=ADR+2
740 ADRS=ADRS+2
750 IF ADRS>=ADRE THEN 3300
760 IF ADR<32768 THEN 780
770 ADR=ADR-65536
780 IF AC=1 THEN 860
790 ON AN3 GOTO 800,820
800 ADI=ADR
810 GOTO 830
820 ADI=ADRS
830 GOSUB 520
840 ADR$=BX$
850 PRINT #P:ADR$;LD$;
860 IF OPT>4 THEN 920
870 CALL PEEK(ADR,B(1),B(2))
880 IF OPT<>3 THEN 930
890 IF SAD<>SU THEN 730
900 GOSUB 4060
910 GOTO 730
920 CALL PEEKV(ADR,B(1),B(2)
)
930 GOSUB 410
940 PRINT #P:BX$;L1$;
950 IF SYT=0 THEN 1010
960 IF RR=1 THEN 1000
970 CAD=ADR
980 GOSUB 3940
990 RETURN
1000 GOSUB 3910
1010 IF OPT>3 THEN 1040
1020 RETURN
1030 REM  ***** DUMP FORMAT
SUBROUTINE *****
1040 AC=1
1050 FOR I=1 TO 2
1060 IF B(I)<32 THEN 1080
1070 IF B(I)<95 THEN 1090
1080 B(I)=46
1090 BP$(I)=CHR$(B(I))
1100 NEXT I
1110 BC$(BL)=BP$(1)&BP$(2)
1120 IF BL=BM THEN 1150
1130 BL=BL+1
1140 GOTO 730
1150 IF P=0 THEN 1170
1160 PRINT #P:L2$;
1170 FOR I=1 TO BM
1180 PRINT #P:BC$(I);
1190 NEXT I
1200 PRINT #P
1210 BL=1
1220 AC=0
1230 GOTO 730
1240 REM ***** REGISTER SUBR
TN *****
1250 RD$=R$&STR$(RD)
1260 ON RIX GOTO 1270,1290,1
310,1450
1270 OP1$=RD$&KO$
1280 RETURN
1290 OP1$=ST$&RD$&KO$
1300 RETURN
1310 RR=1
1320 AC=1
1330 GOSUB 730
1340 IF OI=47 THEN 1350 ELSE
 1380
1350 UF=0
1360 GOSUB 3820
1370 IF UF=1 THEN 1410
1380 IF SF=1 THEN 1400
1390 HX$=KLA$&ZX$
1400 IF RD<>0 THEN 1430
1410 OP1$=HX$&BX$&KO$
1420 RETURN
1430 OP1$=HX$&BX$&KA$&RD$&KZ
$&KO$
1440 RETURN
1450 OP1$=ST$&RD$&PL$&KO$
1460 RETURN
1470 REM   *
```

```
                *
1480 REM  ***** BEGINN HAUPT
PROGRAMM *****
1490 REM   *
                *
1500 RESTORE 380
1510 READ BL,BM,AC,TL,TC,TP,
OPT,SYT,LD$
1520 PRINT "WAEHLE AUS :": :
1530 PRINT "DISASSEMBLER = 1
":
1540 PRINT "DISASS M/SYM = 2
":
1550 PRINT "MEMORY SCAN  = 3
":
1560 PRINT "DUMP.........= 4
":
1570 PRINT "DUMP VDP.....= 5
":
1580 PRINT "PROGRAMMENDE = 6
": : :
1590 INPUT OPT
1600 PRINT
1610 IF OPT<>6 THEN 1630
1620 END
1630 IF OPT=3 THEN 1690
1640 INPUT "DRUCKEN N=0,J=1
":P
1650 IF P=0 THEN 1690
1660 OPEN #P:"RS232",OUTPUT
1670 RESTORE 390
1680 READ BL,BM,AC,TC,TP,TL,
LD$
1690 INPUT "ASM J=1,N=2 ":AN
3
1700 PRINT
1710 IF AN3=2 THEN 1730
1720 GOSUB 4130
1730 CALL CLEAR
1740 PRINT
1750 INPUT "BEGINN,ENDE (HEX
) ":ADR$,ADRE$
1760 PRINT
1770 IF LEN(ADR$)<>4 THEN 17
90
1780 IF LEN(ADRE$)=4 THEN 18
20
1790 PRINT "* 4 CHAR HEXWERT
 EINGEBEN "
1800 CALL SOUND(100,200,3)
1810 GOTO 1750
1820 ON OPT GOTO 1990,1960,1
850,1990,1990
1840 REM **** FIND WITHIN RA
M/ROM ****
1850 INPUT "SUCHBEGRIFF (HEX
) ":AXI$
1860 IF LEN(AXI$)=4 THEN 190
0
1870 PRINT "* SUCHBEGRIFF MU
SS VIERSTELLIG SEIN"
1880 CALL SOUND(100,200,3)
1890 GOTO 1850
1900 GOSUB 620
1910 SU=ADO
1920 AC=1
1940 GOTO 1990
1950 REM *** SETUP INITIAL A
DDRESSES ***
1960 IF SYT=1 THEN 1990
1970 IF UT=1 THEN 1990
1980 GOSUB 3430
1990 AXI$=ADRE$
2000 GOSUB 620
2010 ADRE=ADO
2020 AXI$=ADR$
2030 GOSUB 620
2040 ADR=ADO
2050 ADRS=ADO
2060 RR=0
2070 GOSUB 760
2080 GOTO 2170
2090 REM **** FORMAT ASSEMBL
ER OUTPUT ****
2100 PRINT #P:TAB(TL);LB$;
2110 PRINT #P:TAB(TC);OPC$;T
AB(TP);OP1$;OP2$
2120 RR=0
2130 LB$=LL$
2140 AC=0
2150 GOSUB 730
2160 REM **** FIND OPCODE **
**
2170 OP1$=LL$
2180 OP2$=LL$
2190 HX$=ZX$
2200 KO$=ZK$
2210 B1=INT(B(1)/4)+1
2220 IF B(1)>15 THEN 2250
2230 ON B1 GOTO 2240,2580,26
80,2290
2240 ON B(1)+1 GOTO 2290,229
0,2330,2480
2250 IF B(1)<32 THEN 2740
2260 IF B(1)<64 THEN 2940
2270 GOTO 3080
2280 REM ***** INVALID OPCOD
E IS DATA *****
2290 OPC$=LE1$
2300 PRINT #P:TAB(TC);"DATA"
;L1$;HX$;BX$
2310 GOTO 2120
2320 REM  ***** FORMAT VIII
INSTRUCTIONS *****
2330 OI=UH(B(2))+32
2340 OPC$=OP$(OI)
2350 IF OPC$=LL$ THEN 2290
2360 IF OPC$="LWPI" THEN 238
0
2370 OP1$=R$&STR$(LH(B(2)))&
KO$
2380 AC=1
```

```
2390 RR=1
2400 GOSUB 730
2410 IF B(1)>0 THEN 2440
2420 OP2$=STR$(B(2))
2430 GOTO 2100
2440 IF SF=0 THEN 2460
2450 HX$=LL$
2460 OP2$=HX$&BX$
2470 GOTO 2100
2480 REM ***** FORMAT VII IN
STRUCTIONS *****
2490 IF B(2)>0 THEN 2520
2500 OPC$=OP$(9)
2510 GOTO 2380
2520 F=INT(B(2)/32)
2530 IF B(2)-(F*32)<>0 THEN
2290
2540 OI=F+8
2550 OPC$=OP$(OI)
2560 GOTO 2100
2570 REM ***** FORMAT VI INS
TRUCTIONS *****
2580 H=B(1)+43
2590 OI=H+(UB2*4)
2600 OPC$=OP$(OI)
2610 IF OPC$=LL$ THEN 2290
2620 RD=LH(B(2))
2630 KO$=LL$
2640 RIX=TS+1
2650 GOSUB 1250
2660 GOTO 2100
2670 REM ***** FORMAT V INST
RUCTIONS *****
2680 OI=B(1)+55
2690 OPC$=OP$(OI)
2700 OP1$=R$&STR$(LH(B(2)))&
KO$
2710 OP2$=STR$(UH(B(2)))
2720 GOTO 2100
2730 REM ***** FORMAT II INS
TRUCTIONS *****
2740 OI=B(1)
2750 OPC$=OP$(OI)
2760 IF SEG$(OPC$,1,1)="J" T
HEN 2790
2770 OP1$=STR$(B(2))
2780 GOTO 2100
2790 IF B(2)>127 THEN 2820
2800 ADI=ADR+(B(2)*2)+2
2810 GOTO 2830
2820 ADI=ADR-((255-B(2))*2)
2830 IF SYT=0 THEN 2900
2840 RR=1
2850 CAD=ADI
2860 GOSUB 3930
2870 IF SF=0 THEN 2900
2880 HX$=LL$
2890 GOTO 2910
2900 GOSUB 530
2910 OP1$=HX$&BX$
2920 GOTO 2100
2930 REM ***** FORMAT III/IV
/IX INSTRUCTIONS *****
2940 OI=INT((B(1)-28)/4)
2950 OPC$=OP$(OI)
2960 RD1=(B(1)-(28+(OI*4)))*
4
2970 RD=RD1+UB2
2980 IF LEN(OPC$)=4 THEN 304
0
2990 OP2$=R$&STR$(RD)
3000 RD=LH(B(2))
3010 RIX=TS+1
3020 GOSUB 1250
3030 GOTO 2100
3040 OP2$=STR$(RD)
3050 GOTO 3000
3060 REM
3070 REM  ***** FORMAT I INS
TRUCTIONS *****
3080 OI=UH(B(1))+63
3090 OPC$=OP$(OI)
3100 B1S=B(1)
3110 B2S=B(2)
3120 RIX=TS+1
3130 RD=LH(B(2))
3140 KO$=LL$
3150 GOSUB 1250
3160 OP1S$=OP1$
3170 B(1)=B1S
3180 B(2)=B2S
3190 TD=INT(LH(B(1))/4)
3200 B1L=(LH(B(1))-(TD*4))*4
3210 RIX=TD+1
3220 RD=B1L+UB2
3230 GOSUB 1250
3240 OP2$=OP1$
3250 OP1$=OP1S$&ZK$
3260 GOTO 2100
3270 REM  *
   *
3280 REM ***** ENDE PROGM **
***
3290 REM  *
   *
3300 PRINT #P
3310 PRINT
3320 INPUT "WEITER ? J/N ":A
$
3330 IF A$="N" THEN 3400
3340 IF P=0 THEN 3370
3350 RESTORE 390
3360 GOTO 3380
3370 RESTORE 380
3380 READ BL,BM,AC
3390 GOTO 1740
3400 IF P=0 THEN 1500
3410 CLOSE #P
3420 GOTO 1500
3430 REM **** EINLESEN UTILI
TY TABLE *****
3440 UADR=28438
```

```
3450 PRINT :"UTLTAB... ";
3460 FOR I=1 TO 28
3470 CALL PEEK(UADR,N(1),N(2
),N(3),N(4),N(5),N(6),B(1),B
(2))
3480 FOR J=1 TO 6
3490 NA$(J)=CHR$(N(J))
3500 NEXT J
3510 UNAM$(I)=NA$(1)&NA$(2)&
NA$(3)&NA$(4)&NA$(5)&NA$(6)
3520 UAD(I)=SAD
3530 UADR=UADR+8
3540 NEXT I
3550 UT=1
3560 PRINT "EINGELESEN": :
3570 IF OPT=2 THEN 3600
3580 RETURN
3590 REM ***** EINLESEN SYMB
OL TABLE *****
3600 SYCA=29072
3610 SYTA=31960
3620 SYT=1
3630 CALL PEEK(SYCA,B(1),B(2
))
3640 IF SAD>1 THEN 3670
3650 SYT=0
3660 RETURN
3670 SYC=SAD
3680 FOR V=1 TO SYC
3690 CALL PEEK(SYTA,N(1),N(2
),B(1),B(2))
3700 FOR J=1 TO 2
3710 S$(J)=CHR$(N(J))
3720 NEXT J
3730 SY$(V)=S$(1)&S$(2)
3740 SYA(V)=SAD
3750 GOSUB 420
3760 PRINT #P:TAB(6);SY$(V);
L1$;"EQU";L1$;ZX$;BX$
3770 SYTA=SYTA+4
3780 NEXT V
3790 PRINT #P
3800 RETURN
3810 REM **** SCAN UTLTAB **
**
3820 IF UT=0 THEN 3860
3830 FOR I=1 TO 28
3840 IF SAD=UAD(I)THEN 3870
3850 NEXT I
3860 RETURN
3870 BX$=UNAM$(I)
3880 HX$=KLA$
3890 UF=1
3900 RETURN
3910 REM **** SCAN SYMTAB **
**
3920 CAD=SAD
3930 SF=0
3940 FOR I=1 TO SYC
3950 IF CAD=SYA(I)THEN 3980
3960 NEXT I
3970 RETURN
3980 SF=1
3990 IF RR=1 THEN 4020
4000 LB$=SY$(I)
4010 RETURN
4020 BX$=SY$(I)
4030 RR=0
4040 GOTO 3880
4050 REM ***** PRINT GEF.SUC
HBEGIFF ****
4060 ADI=ADR
4070 GOSUB 420
4080 SU$=BX$
4090 GOSUB 530
4100 PRINT #P:BX$;L1$;ZX$;SU
$
4110 GOTO 730
4120 REM ***** TEST U LADEN
ASM PROG *****
4130 DATA 84,79,88,36,32,32,
127,102
4140 PAD=32760
4150 RESTORE 4130
4160 FOR I=0 TO 3
4170 READ A
4180 CALL PEEK(PAD+I,NM)
4190 IF NM<>A THEN 4220
4200 NEXT I
4210 RETURN
4220 PRINT "** TOX$ PGM NICH
T GELADEN. **"
4230 PRINT "LADEN BELEGT >7F
60 BIS >7FF0 "
4240 PRINT "UND REF/DEF TABL
E >7FF8 ": :
4250 INPUT "OK? J/N ":AN$
4260 IF AN$="N" THEN 4270 EL
SE 4290
4270 AN3=2
4280 RETURN
4290 DATA 194,139,4,192,2,1,
0,1,2,5,127,96,2,6,4,0,221,7
0,6,160
4300 DATA 127,180,194,4,5,12
9,6,160,127,180,2,136,0,1,22
,5,6,160,127,198
4310 DATA 10,68,6,160,127,19
8,10,132,6,160,127,198,10,19
6,6,160,127,198,5,129
4320 DATA 2,136,0,1,19,29,2,
129,0,4,19,26,6,160,127,180,
16,240,4,32
4330 DATA 96,68,4,32,96,28,1
8,0,193,32,131,74,193,196,4,
91,8,132,209,0
4340 DATA 8,68,2,132,0,10,17
,2,2,36,0,7,2,36,0,48,10,132
,221,68
4350 DATA 193,7,4,91,2,2,127
,96,4,32,96,72,4,90
4360 LP=7
```

```
4370 RESTORE 4130
4380 FOR I=0 TO LP
4390 READ BY
4400 CALL LOAD(PAD+I,BY)
4410 NEXT I
4420 IF LP=133 THEN 4470
4430 PAD=32614
4440 LP=133
4450 RESTORE 4290
4460 GOTO 4380
4470 PRINT
4472 CALL LOAD(28702,127,248
)
4480 RETURN
```




```
10 ! **********************
11 ! *                    *
12 ! *       RESET        *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *  Peter Vorwerk     *
17 ! *                    *
18 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *  32-K Erweiterung  *
23 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     1599 Bytes     *
28 ! *                    *
29 ! **********************
```

```
30 !
100 RESTORE 350 :: FOR I=1 T
O 10 :: READ CH(I):: NEXT I
110 RESTORE :: FOR I=1 TO 10
 :: SUM=0 :: FOR J=1 TO 16 :
: READ A :: SUM=SUM+A :: NEX
T J
120 IF SUM<>CH(I)THEN PRINT
"* FEHLER IN ZEILE";I*10+130
 :: CALL SOUND(100,110,0)::
STOP
130 NEXT I
140 PRINT " ALLES O.K.":" JE
TZT WIRD GEPOKED"
150 RESTORE
160 ADR=15398
170 CALL INIT
180 READ A :: IF A<>-1 THEN
CALL LOAD(ADR,A):: ADR=ADR+1
 :: GOTO 180
190 CALL LOAD(8194,60,186,63
,248)
200 FOR I=16376 TO 16383 ::
READ A :: CALL LOAD(I,A):: N
EXT I
210 DATA 0,99,2,224,60,0,2,1
,255,216,6,65,6,65,216,49
220 DATA 60,36,216,49,60,34,
216,49,60,32,152,32,60,36,60
,38
230 DATA 22,244,152,32,60,34
,60,38,22,240,152,32,60,32,6
0,38
240 DATA 22,236,200,1,60,36,
2,33,0,2,216,49,60,34,216,49
250 DATA 60,35,2,33,0,2,216,
49,60,32,216,49,60,33,152,32
260 DATA 60,32,60,36,17,13,1
52,32,60,32,60,34,21,242,17,
4
270 DATA 152,32,60,33,60,35,
20,237,200,32,60,32,60,34,16
,233
280 DATA 6,96,60,35,200,32,6
0,36,131,49,200,32,60,34,128
,51
290 DATA 216,32,60,39,131,35
,216,32,60,38,131,124,2,224,
131,224
300 DATA 4,96,0,112,255,0,25
5,0,255,0,255,0,255,0,255,0
310 DATA -1
320 REM NAMENSTABELLE:
330 DATA 82,69,83,69,84,32,6
0,40,0,0,0,0,0,0,0,0
340 REM PRUEFSUMMEN
350 DATA 1266,1190,1278,1216
,1031,872,1296,1210,1695,174
2
```

```
10 REM*********************
11 REM*                   *
12 REM*  ERDBEER-PAUL     *
13 REM*  Copyright  by    *
14 REM*Gunther  Kraemling*
15 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                   *
23 REM* Speicherbelegung *
24 REM*    6194 Bytes     *
25 REM*********************
26 REM
200 CALL CLEAR
210 CALL CHAR(33,"FFFFFFFFFF
FFFFFF")
220 CALL CHAR(34,"FFFFFF")
230 CALL CHAR(40,"FFFFFFFFFF
FFFFFF")
240 CALL CHAR(58,"7E5A7E18FF
182442")
250 CALL CHAR(59,"00007E5AFF
99A5A5")
260 CALL CHAR(64,"060C9C4830
080808")
270 CALL CHAR(72,"0C1E1E0EE4
BFFF0E")
280 CALL CHAR(80,"3C243C243C
243C24")
290 CALL CHAR(88,"00000C0201
2D7EFE")
300 CALL CHAR(89,"187E181818
3C7EFF")
310 CALL CHAR(96,"3F3F070F1F
3F7FFF")
320 CALL CHAR(97,"FCFCE0F0F8
FCFEFF")
330 CALL CHAR(98,"FFFFFFFFFF
7F3F1F")
340 CALL CHAR(102,"FFFFFFFFF
FFEFCF8")
350 CALL CHAR(106,"0000004CF
C40")
360 CALL CHAR(107,"000000023
F02")
370 CALL CHAR(120,"0E1F3F3F1
E0E")
380 CALL CHAR(128,"0000387E7
E7E3C18")
390 CALL SCREEN(7)
400 CALL COLOR(1,3,1)
410 CALL COLOR(2,14,1)
420 CALL COLOR(3,2,1)
430 CALL COLOR(4,2,1)
440 CALL COLOR(5,3,1)
450 CALL COLOR(6,12,1)
460 CALL COLOR(7,11,1)
470 CALL COLOR(8,15,1)
480 CALL COLOR(9,6,1)
490 CALL COLOR(10,6,1)
500 CALL COLOR(11,6,1)
510 CALL COLOR(12,9,1)
520 CALL COLOR(13,9,16)
530 CALL HCHAR(3,1,40,32)
540 CALL HCHAR(20,1,33,32)
550 CALL HCHAR(8,1,40,16)
560 CALL VCHAR(9,16,40,9)
570 CALL HCHAR(10,17,34,3)
580 CALL HCHAR(10,21,34,7)
590 CALL HCHAR(10,29,34,4)
600 CALL HCHAR(15,1,34,4)
610 CALL HCHAR(15,7,34,7)
620 CALL HCHAR(15,15,34)
630 CALL HCHAR(15,17,34)
640 CALL HCHAR(15,19,34,11)
650 CALL HCHAR(15,31,34,2)
660 CALL HCHAR(18,5,34,2)
670 CALL VCHAR(10,20,80,4)
680 CALL VCHAR(10,28,80,4)
690 CALL VCHAR(15,14,80,4)
700 CALL VCHAR(15,18,80,4)
710 CALL VCHAR(15,30,80,4)
720 CALL VCHAR(16,5,96)
730 CALL VCHAR(16,6,97)
740 CALL VCHAR(17,5,98)
750 CALL VCHAR(17,6,102)
760 HP=0
770 B=24
780 C=32
790 X=19
800 Y=3
810 A=7
820 P=0
830 D=4
840 E=24
850 F=5
860 R=24
861 U=0
870 FOR I=19 TO 31 STEP 3
880 CALL HCHAR(8,I,64)
890 CALL HCHAR(8,I-1,120)
900 NEXT I
910 CALL HCHAR(13,19,64)
920 CALL HCHAR(13,18,120)
930 CALL HCHAR(13,23,64)
940 CALL HCHAR(13,22,120)
950 CALL HCHAR(13,26,64)
960 CALL HCHAR(13,25,120)
970 CALL HCHAR(13,31,64)
980 CALL HCHAR(13,30,120)
990 FOR I=21 TO 27 STEP 3
1000 CALL HCHAR(18,I,64)
1010 CALL HCHAR(18,I-1,120)
1020 NEXT I
1030 CALL HCHAR(14,A,59)
1040 CALL HCHAR(14,B,59)
1050 CALL HCHAR(9,R,59)
1060 CALL HCHAR(19,C,88)
1070 CALL HCHAR(D,E,72)
1080 GOSUB 2870
1090 GOSUB 2910
```

```
1100 GOSUB 2950
1110 CALL HCHAR(X,Y,58)
1120 CALL KEY(0,K,S)
1130 IF S=0 THEN 1570
1140 IF K<>68 THEN 1220
1150 IF Y=32 THEN 1590
1160 CALL GCHAR(X,Y+1,M)
1170 IF M=0 THEN 1200
1180 IF M=40 THEN 1590
1190 CALL HCHAR(X,Y,32)
1200 Y=Y+1
1210 GOTO 1570
1220 IF K<>83 THEN 1310
1230 IF Y=1 THEN 1590
1240 IF (Y=7)*(X=14)THEN 143
0
1250 CALL GCHAR(X,Y-1,M)
1260 IF M=0 THEN 1260
1270 IF M=40 THEN 1590
1280 CALL HCHAR(X,Y,32)
1290 Y=Y-1
1300 GOTO 1570
1310 IF K<>69 THEN 1370
1320 CALL GCHAR(X-1,Y,M)
1330 IF M<>80 THEN 1590
1340 CALL HCHAR(X,Y,32)
1350 X=X-5
1360 GOTO 1570
1370 IF K<>88 THEN 1590
1380 CALL GCHAR(X+1,Y,M)
1390 IF M<>80 THEN 1590
1400 CALL HCHAR(X,Y,32)
1410 X=X+5
1420 GOTO 1570
1430 IF N=0 THEN 1570
1440 FOR I=1 TO 3
1450 CALL SOUND(150,-I,6)
1460 NEXT I
1470 P=P+30
1480 GOSUB 2870
1490 F=F+6
1500 GOSUB 2910
1510 N=0
1511 U=U+1
1512 IF U=12 THEN 861
1520 CALL HCHAR(21,27,32)
1530 CALL HCHAR(14,6,120)
1540 CALL HCHAR(14,6,32)
1550 CALL HCHAR(15,5,120)
1560 CALL HCHAR(15,5,32)
1570 CALL HCHAR(X,Y,58)
1580 CALL SOUND(20,600,6)
1590 IF K<>81 THEN 1660
1600 CALL GCHAR(X-1,Y,M)
1610 IF M<>120 THEN 1660
1620 CALL HCHAR(X-1,Y,32)
1630 CALL HCHAR(21,27,128)
1640 N=1
1650 CALL SOUND(50,2000,5)
1660 IF K<>46 THEN 1800
1670 IF Y=32 THEN 2120
1680 CALL GCHAR(X,Y+1,M)
1690 IF M=40 THEN 2120
1700 IF M=32 THEN 1760
1710 CALL HCHAR(X,Y+1,106)
1720 F=F-1
1730 CALL HCHAR(X,Y+1,32)
1740 GOTO 1930
1750 F=F-1
1760 CALL HCHAR(X,Y+1,106)
1770 F=F-1
1780 CALL HCHAR(X,Y+1,32)
1790 GOTO 2100
1800 IF K<>44 THEN 2120
1810 IF Y=1 THEN 2120
1820 CALL GCHAR(X,Y-1,M)
1830 IF M=40 THEN 2120
1840 IF M=32 THEN 1890
1850 CALL HCHAR(X,Y-1,107)
1860 F=F-1
1870 CALL HCHAR(X,Y-1,32)
1880 GOTO 1930
1890 CALL HCHAR(X,Y-1,107)
1900 F=F-1
1910 CALL HCHAR(X,Y-1,32)
1920 GOTO 2100
1930 IF X<>19 THEN 1970
1940 C=32
1950 CALL HCHAR(19,C,88)
1960 GOTO 2060
1970 IF X<>9 THEN 2010
1980 R=32
1990 CALL HCHAR(9,R,59)
2000 GOTO 2060
2010 IF Y<16 THEN 2050
2020 B=32
2030 CALL HCHAR(14,B,59)
2040 GOTO 2060
2050 A=1
2060 CALL HCHAR(14,A,59)
2070 CALL SOUND(150,110,6)
2080 P=P+15
2090 GOSUB 2870
2100 GOSUB 2910
2110 IF F=0 THEN 2650
2120 IF X<>19 THEN 2220
2130 IF Y<C THEN 2170
2140 C=C+1
2150 CALL HCHAR(19,C-1,32)
2160 GOTO 2190
2170 C=C-1
2180 CALL HCHAR(19,C+1,32)
2190 CALL HCHAR(19,C,88)
2200 IF C=Y THEN 2650
2210 GOTO 1120
2220 IF X<>9 THEN 2500
2230 IF Y>R THEN 2270
2240 R=R-1
2250 CALL HCHAR(9,R+1,32)
2260 GOTO 2290
2270 R=R+1
```

```
2280 CALL HCHAR(9,R-1,32)
2290 CALL HCHAR(9,R,59)
2300 IF R=Y THEN 2650
2310 IF Y>E THEN 2380
2320 CALL GCHAR(D+1,E-1,M)
2330 IF (M=32)+(M=58)THEN 23
40 ELSE 2440
2340 D=D+1
2350 E=E-1
2360 CALL HCHAR(D-1,E+1,32)
2370 GOTO 2470
2380 CALL GCHAR(D+1,E+1,M)
2390 IF (M=32)+(M=58)THEN 24
00 ELSE 2440
2400 D=D+1
2410 E=E+1
2420 CALL HCHAR(D-1,E-1,32)
2430 GOTO 2470
2440 CALL HCHAR(D,E,32)
2450 D=4
2460 C=24
2470 CALL HCHAR(D,E,72)
2480 IF (E=Y)*(D=X)THEN 2650
2490 GOTO 1120
2500 IF Y<16 THEN 2600
2510 IF Y<B THEN 2550
2520 B=B+1
2530 CALL HCHAR(14,B-1,32)
2540 GOTO 2570
2550 B=B-1
2560 CALL HCHAR(14,B+1,32)
2570 CALL HCHAR(14,B,59)
2580 IF B=Y THEN 2650
2590 GOTO 1120
2600 A=A+1
2610 CALL HCHAR(14,A-1,32)
2620 CALL HCHAR(14,A,59)
2630 IF A=Y THEN 2650
2640 GOTO 1120
2650 CALL HCHAR(X,Y,89)
2660 FOR I=70 TO 100
2670 CALL SOUND(10,210-I,5)
2680 NEXT I
2690 FOR I=1 TO 500
2700 CALL KEY(0,K,S)
2710 IF S<>0 THEN 2750
2720 NEXT I
2730 CALL CLEAR
2740 END
2750 CALL HCHAR(X,Y,32)
2760 CALL HCHAR(14,A,32)
2770 CALL HCHAR(14,B,32)
2780 CALL HCHAR(9,R,32)
2790 CALL HCHAR(19,C,32)
2800 CALL HCHAR(D,E,32)
2810 CALL HCHAR(21,27,32)
2820 CALL HCHAR(23,13,32,5)
2830 CALL HCHAR(24,13,32,5)
2840 IF P<=HP THEN 770
2850 HP=P
2860 GOTO 770
2870 U$="score>"&STR$(P)&"
     "
2880 V=23
2890 W=6
2900 GOTO 2980
2910 U$="swords>"&STR$(F)&"
     "
2920 V=24
2930 W=5
2940 GOTO 2980
2950 U$="highscore>"&STR$(HP
)&""
2960 V=22
2970 W=2
2980 FOR I=1 TO LEN(U$)
2990 Z=ASC(SEG$(U$,I,1))
3000 CALL HCHAR(V,W,Z)
3010 W=W+1
3020 NEXT I
3030 RETURN
```

BÖRSE

Verkaufe Module, Disketten, Kassetten für TI. Pochner, Brunhildstr. 31, 5300 Bonn 2

Achtung: TI-User!! Suche Kontakt zu anderen Usern zum Austasuch von Tips und Programmen. Suche auch Sports. und Flugsimulationen in X Basic. Preis auf Vereinbarung. Ingo Lachner, Wil.-Raabe-Str. 29, 851 Fürth/Bay. (PS ohne 32K)

Verkaufe TI-99/4A+Ext. Basic + Recorder + Joystick + Joy. Adapt. + 100 Progr. (z.B. Q-Bert) + 3 Bücher + Alligator Mix + Ect. Ba. Lernprogramm + 4 TI-Revue + 1 TI-Speciai für nur 500,– Tel. 09741/36 54

Original TI-Kass. (Adventure) Voodoo Castle – Adventureland – The golden Voyage – Thost Town – Savage Island Series – je DM 24,– ExB: Sortierprogramm SORT 500 in deutsch Orig. Kass. DM 25,–. Krummrey, Lipschitzallee 71, 1000 Berlin 47

Verk. Buchungsjournal 150,–, TI-Basic Tutor (Cass.) 10,–, Rechnungsstellung, Lagerverwaltung je 80,– Modul Statistik 40,–, Datenverw. + Anal. 75,–, Vokabeltrainer (Cass.) 10,– Connect Four (Mod.) 20,–. Tel. 0711/71 92 67, R. Unruh

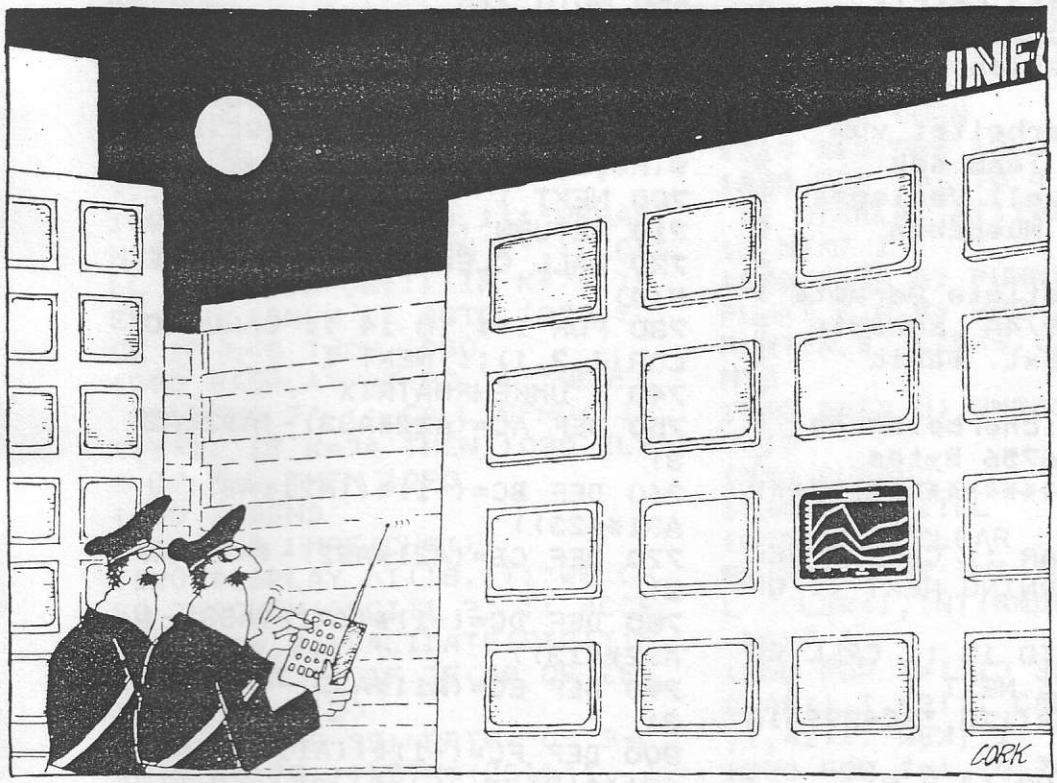
---

Verk. 32k-Ext. durchgef. Bus + Centron, Sprachsynthes. Extended Basic Modul, versch. Spiel Module, Literatur. Angebote ab 20 Uhr. Tel. 0211/733 26 76

Tausche Centronics-Interface gegen RS232-Interface. Sind Sie interesiert? Dann rufen Sie mich doch an: 0201/53 06 87

TI 99 4A komplettes Symste: Box, Disk, 32K, RS, Speech, Drucker, XB, E/A, MM, TE II und vieles mehr! Software + Bücher satt! VB 2800,– 09321–4636

Verkaufe: TI 99/4A + Exbasic + TI Joystics + 32 K-CMOS Ramkarte + Rec. Kabel + deut. Ex-Basicanleitung. Preis: VS. Tel. 02871/66 38



DIE HAND WERKER KOMMEN

Soft-Plot

Soft-Plot dient zur schnellen Erstellung einfarbiger Graphiken und mehrere Graphik-Funktionen erleichtern dabei die Arbeit.

1. Radier-Funktionen: löscht jeweils ein 4*4 Dots großes Quadrat
2. Vergrößern: 4-fache Vergrößerung des Zeichens.
3. Kreis-Funktion: Plottet Kreise mit beliebigen Radien.
4. Linien-Funktion: Plottet eine Linie zwischen Anfangs- und Endpunkt. Gleichmäßige Punkteverteilung.
5. Char-Funktion: Kopieren eines Zeichens, um freie Zeichen zu sparen.
6. Lösch-Funktion: Löscht den Bildschirm und setzt alle Zeichen wieder auf „0".
7. Ausdruck: Programm Ende. Ausdruck des Hexcodes sowie der Position D. Zeichen.

Ist der Zeichenvorrat erschöpft, kann man in den gesetzten Zeichen nachplotten.

Das Programm erklärt sich im wesentlichen selbst – will man die Erklärung jedoch weglassen, sind folgende Zeilen zu löschen:

> 220
> 240
> 1550 bis Ende

> 170 – Einleitung.
> 300 – Bildschirm Aufbau
> 450 – Option: Vergrößern
> 510 – Option: Radieren
> 610 – Option: Linien ziehen.
> 630 – Option: Kreise zeichnen
> 730 – Option: Zeichen plazieren
> 760 – Unterprogramm Linie
> 900 – Unterprogramm Plotten
> 1170 – Steuerung des Zeichenstiftes.
> 1230 – Wahl der Optionen
> 1420 – Anzeige der Anzahl freier Zeichen
> 1490 – Ausdruck der Zeichen und Programmende.
> 1550 – Erklärung des Programms
> 1710 – Textdaten der Erklärung

3x3 Matrix

Habt Ihr schon mal neidisch auf andere Computersysteme geschaut, in deren Betriebssprache sogar Matrizenbefehle integriert sind?

Für einen einfachen Befehl in der Textverarbeitung, z.B. 100 MAT A=RxS, müßt Ihr in TI- oder Ext.-Basic schon ein ganzes Programm schreiben, andere Befehle wie MAT READ, MAT INPUT usw. sind ebenfalls nur sehr umständlich in die Sprache des TI99 zu übersetzen. Mit diesem Programm wollen wir Euch zeigen, wie Ihr Euch MAT-Befehle für Euren TI schreiben könnt. Es kann 3 MAT-Statements in Verbindung mit einer 3x3 Matrix realisieren.
1.) MAT A=BxC, 2. MAT A=B+C 3.) MAT A=INV (B).

In die Felder der Matrizen dürft Ihr allerdings nur bis zu 3-stellige Zahlen (incl. Nachkommatrennzeichen) eingeben. Dabei müßt Ihr darauf achten, daß das Ergebnis höchstens vier Stellen besitzt, sonst liefert Euch der Computer nur Sternchen.

Ein Menü führt Euch durch das Programm.

Habt Ihr etwas Ahnung von Matrixoperationen, so seht Euch dieses Listing genau an. Wenn Ihr das Prinzip verstanden habt, dürfte es Euch eigentlich nicht schwer fallen, das Programm beliebig zu erweitern. Neidische Blicke auf andere Systeme gehören damit der Vergangenheit an.

Chr. Villwock

```
100 ! **********************
110 ! *                    *
120 ! *     3X3 Matrix     *
130 ! *    Copyright by    *
140 ! * Christian Villwock *
150 ! *                    *
160 ! *   Bearbeitet vom   *
170 ! *       Team des     *
180 ! *  Aktuell Verlages  *
190 ! *       Muenchen     *
200 ! *                    *
210 ! * Benoetigte Geraete *
220 ! *  TI99/4A  Konsole  *
230 ! *     Ext. Basic     *
240 ! *                    *
270 ! *  Speicherbelegung  *
280 ! *     6756 Bytes     *
290 ! **********************
295 !
300 CALL CLEAR :: CALL SCREE
N(2):: ON WARNING NEXT :: ON
 BREAK NEXT
310 FOR I=1 TO 13 :: CALL CO
LOR(I,16,1):: NEXT I
320 CALL CHAR(120,"3C4299A1A
199423C")
330 CALL COLOR(14,9,9)
340 R=3
350 M$="x 1984"
360 GOSUB 660
370 R=5
380 M$="PROGRAMM VON"
390 GOSUB 660
400 R=7
410 M$="CHRISTIAN VILLWOCK"
420 GOSUB 660
430 R=9
440 M$="--------------------
-"
450 GOSUB 660
460 R=10
470 M$="3 * 3 M A T R I Z E
N"
480 GOSUB 660
490 R=11
500 M$="--------------------
-"
510 GOSUB 660
520 R=13
530 M$="FUER"
540 GOSUB 660
550 R=15
560 M$="TI REVUE LESER"
570 GOSUB 660
580 R=17
590 M$="UND "
600 GOSUB 660
610 R=19
620 M$="ALLE ANDEREN"
630 GOSUB 660
640 FOR I=1 TO 1000 :: NEXT
I
650 GOTO 720
660 C=16-LEN(M$)/2
670 FOR I=1 TO LEN(M$)
680 CALL HCHAR(R,C+I,136)
690 CALL HCHAR(R,C+I,ASC(SEG
$(M$,I,1)))
700 NEXT I
710 RETURN
720 CALL CLEAR :: CALL SCREE
N(4)
730 FOR I=1 TO 14 :: CALL CO
LOR(I,2,1):: NEXT I
740 ! UMKEHRMATRIX
750 DEF AC=(A22*A33)-(A32*A2
3)
760 DEF BC=(-1)*((A21*A33)-(
A31*A23))
770 DEF CB=(A21*A32)-(A31*A2
2)
780 DEF DC=(-1)*((A12*A33)-(
A32*A13))
790 DEF EC=(A11*A33)-(A31*A1
3)
800 DEF FC=(-1)*((A11*A32)-(
A31*A12))
810 DEF GC=(A12*A23)-(A22*A1
3)
820 DEF HC=(-1)*((A11*A23)-(
A21*A13))
830 DEF IC=(A11*A22)-(A21*A1
2)
840 DEF DET=(A11*A22*A33)+(A
12*A23*A31)+(A21*A32*A13)-(A
13*A22*A31)-(A11*A23*A32)-(A
33*A12*A21)
850 ! MULTIPLIKATION
860 DEF AA=(A11*B11)+(A12*B2
1)+(A13*B31)
870 DEF BB=(A11*B12)+(A12*B2
2)+(A13*B32)
880 DEF CC=(A11*B13)+(A12*B2
3)+(A13*B33)
890 DEF DD=(A21*B11)+(A22*B2
1)+(A23*B31)
900 DEF EE=(A21*B12)+(A22*B2
2)+(A23*B32)
910 DEF FF=(A21*B13)+(A22*B2
3)+(A23*B33)
920 DEF GG=(A31*B11)+(A32*B2
1)+(A33*B31)
930 DEF HH=(A31*B12)+(A32*B2
2)+(A33*B32)
940 DEF II=(A31*B13)+(A32*B2
3)+(A33*B33)
950 ! ADDITION
960 DEF AB=A11+B11
970 DEF BA=A12+B12
980 DEF CA=A13+B13
990 DEF DA=A21+B21
```

```
1000 DEF EA=A22+B22
1010 DEF FA=A23+B23
1020 DEF GA=A31+B31
1030 DEF HA=A32+B32
1040 DEF IA=A33+B33
1050 ! ZEICHEN FUER TABELLE
1060 CALL CHAR(100,"000000FF
FF",101,"030303030303FFFF",1
02,"000000000000FFFF",104,"0
303037F7F030303",103,RPT$("0
3",8))
1070 CALL CHAR(105,"18181818
1818FFFF",106,"181818FFFF181
818",107,RPT$("18",8))
1080 CALL COLOR(9,2,1,10,2,1
)
1090 ! MENUE
1100 CALL CLEAR :: CALL SCRE
EN(4)
1110 DISPLAY AT(5,8):"3*3 MA
TRIZEN"
1120 DISPLAY AT(7,6):"------
-----------"
1130 DISPLAY AT(9,7):"PROGRA
MMWAHL :"
1140 DISPLAY AT(11,7):"1 =>
MULTIPLIKATION"
1150 DISPLAY AT(13,7):"2 =>
ADDITION"
1160 DISPLAY AT(15,7):"3 =>
UMKEHRMATRIX"
1170 DISPLAY AT(17,7):"4 =>
ENDE"
1180 DISPLAY AT(20,7):"BITTE
 WAEHLEN SIE :"
1190 ACCEPT AT(22,7)VALIDATE
("1234")BEEP SIZE(1):WAHL
1200 IF WAHL=1 THEN 1240 ELS
E IF WAHL=2 THEN 1240 ELSE I
F WAHL=3 THEN 1240 ELSE IF W
AHL=4 THEN 2590
1210 GOTO 1190
1220 ! ABFRAGE M1
1230 XYZ=1
1240 GOSUB 2040
1250 CALL SCREEN(6)
1260 DISPLAY AT(3,7):"1g..1.
.k..2..k..3..k"
1270 ACCEPT AT(6,10)VALIDATE
(NUMERIC)BEEP SIZE(3):A11
1280 ACCEPT AT(6,16)VALIDATE
(NUMERIC)SIZE(3):A12
1290 ACCEPT AT(6,22)VALIDATE
(NUMERIC)SIZE(3):A13
1300 ACCEPT AT(9,10)VALIDATE
(NUMERIC)BEEP SIZE(3):A21
1310 ACCEPT AT(9,16)VALIDATE
(NUMERIC)SIZE(3):A22
1320 ACCEPT AT(9,22)VALIDATE
(NUMERIC)SIZE(3):A23
1330 ACCEPT AT(12,10)VALIDAT
E(NUMERIC)BEEP SIZE(3):A31
1340 ACCEPT AT(12,16)VALIDAT
E(NUMERIC)SIZE(3):A32
1350 ACCEPT AT(12,22)VALIDAT
E(NUMERIC)SIZE(3):A33
1360 XYZ=1
1370 GOSUB 2620
1380 IF WAHL=3 THEN 2180
1390 CALL CLEAR
1400 ! ABFRAGE M2
1410 XYZ=2
1420 GOSUB 2040
1430 DISPLAY AT(3,7):"2g..1.
.k..2..k..3..k"
1440 ACCEPT AT(6,10)VALIDATE
(NUMERIC)BEEP SIZE(3):B11
1450 ACCEPT AT(6,16)VALIDATE
(NUMERIC)SIZE(3):B12
1460 ACCEPT AT(6,22)VALIDATE
(NUMERIC)SIZE(3):B13
1470 ACCEPT AT(9,10)VALIDATE
(NUMERIC)BEEP SIZE(3):B21
1480 ACCEPT AT(9,16)VALIDATE
(NUMERIC)SIZE(3):B22
1490 ACCEPT AT(9,22)VALIDATE
(NUMERIC)SIZE(3):B23
1500 ACCEPT AT(12,10)VALIDAT
E(NUMERIC)BEEP SIZE(3):B31
1510 ACCEPT AT(12,16)VALIDAT
E(NUMERIC)SIZE(3):B32
1520 ACCEPT AT(12,22)VALIDAT
E(NUMERIC)SIZE(3):B33
1530 IF WAHL=1 OR WAHL=2 THE
N GOSUB 2620
1540 IF WAHL=1 THEN 1560 ELS
E IF WAHL=2 THEN 1740
1550 !  MULTIPLIKATION
1560 GOSUB 2040
1570 DISPLAY AT(3,5):"1*2g..
1..k..2..k..3..k"
1580 DISPLAY AT(15,8):"MULTI
PLIKATION !"
1590 DISPLAY AT(6,9):USING "
####":AA
1600 DISPLAY AT(6,15):USING
"####":BB
1610 DISPLAY AT(6,21):USING
"####":BB
1620 DISPLAY AT(9,9):USING "
####":DD
1630 DISPLAY AT(9,15):USING
"####":EE
1640 DISPLAY AT(9,21):USING
"####":FF
1650 DISPLAY AT(12,9):USING
"####":GG
1660 DISPLAY AT(12,15):USING
 "####":HH
1670 DISPLAY AT(12,21):USING
 "####":II
```

```
1680 GOSUB 2690
1690 GOSUB 1900
1700 CALL KEY(0,K,S):: IF S=
0 THEN 1700 ELSE 1080
1710 GOTO 1700
1720 ! ADDITION
1730 GOSUB 2040
1740 DISPLAY AT(6,9):USING "
####":AB
1750 DISPLAY AT(3,5):"1+2g..
1..k..2..k..3..k"
1760 DISPLAY AT(15,11):"ADDI
TION !"
1770 DISPLAY AT(6,15):USING
"####":BA
1780 DISPLAY AT(6,21):USING
"####":CA
1790 DISPLAY AT(9,9):USING "
####":DA
1800 DISPLAY AT(9,15):USING
"####":EA
1810 DISPLAY AT(9,21):USING
"####":FA
1820 DISPLAY AT(12,9):USING
"####":GA
1830 DISPLAY AT(12,15):USING
 "####":HA
1840 DISPLAY AT(12,21):USING
 "####":IA
1850 GOSUB 2690
1860 GOSUB 1900
1870 CALL KEY(0,K,S):: IF S=
0 THEN 1870 ELSE 1080
1880 GOTO 1870
1890 ! DARSTELLUNG M1 U. M2
1900 DISPLAY AT(17,2):A11 ::
 DISPLAY AT(17,6):A12
1910 DISPLAY AT(17,10):A13 :
: DISPLAY AT(19,2):A21
1920 DISPLAY AT(19,6):A22 ::
 DISPLAY AT(19,10):A23
1930 DISPLAY AT(21,2):A31 ::
 DISPLAY AT(21,6):A32
1940 DISPLAY AT(21,10):A33 :
: DISPLAY AT(17,17):B11
1950 DISPLAY AT(17,21):B12 :
: DISPLAY AT(17,25):B13
1960 DISPLAY AT(19,17):B21 :
: DISPLAY AT(19,21):B22
1970 DISPLAY AT(19,25):B23 :
: DISPLAY AT(21,17):B31
1980 DISPLAY AT(21,21):B32 :
: DISPLAY AT(21,25):B33
1990 IF WAHL=1 THEN CALL HCH
AR(19,17,42)
2000 IF WAHL=2 THEN CALL HCH
AR(19,17,43)
2010 DISPLAY AT(24,2):"<< TA
STE DRUECKEN >>"
2020 ! ZEICHNEN DER TABELLE
2030 RETURN
2040 CALL CLEAR
2050 DISPLAY AT(3,8):"g..1..
...2.....3"
2060 DISPLAY AT(4,7):"fefffff
fifffffiffffffi"
2070 DISPLAY AT(5,8):"g.....
k.....k.....k"
2080 DISPLAY AT(6,7):"1h....
.k.....k.....k"
2090 DISPLAY AT(7,8):"gddddd
jddddjddddddj"
2100 DISPLAY AT(8,8):"g.....
k.....k.....k"
2110 DISPLAY AT(9,7):"2h....
.k.....k.....k"
2120 DISPLAY AT(10,8):"gdddd
djddddjddddddj"
2130 DISPLAY AT(11,8):"g....
.k.....k.....k"
2140 DISPLAY AT(12,7):"3h...
..k.....k.....k"
2150 DISPLAY AT(13,8):"gdddd
djddddjddddddj"
2160 RETURN
2170 ! UMKEHRMATRIX
2180 GOSUB 2040
2190 DISPLAY AT(15,7):"UMKEH
RMATRIX !"
2200 DISPLAY AT(17,6):"DIE D
ETERMINANTE LAUTET :"
2210 DISPLAY AT(19,10):DET
2220 IF DET=0 THEN 2560
2230 DISPLAY AT(3,7):"Tg..1.
.k..2..k..3..k"
2240 DISPLAY AT(6,9):USING "
####":AC
2250 DISPLAY AT(6,15):USING
"####":BC
2260 DISPLAY AT(6,21):USING
"####":CB
2270 DISPLAY AT(9,9):USING "
####":DC
2280 DISPLAY AT(9,15):USING
"####":EC
2290 DISPLAY AT(9,21):USING
"####":FC
2300 DISPLAY AT(12,9):USING
"####":GC
2310 DISPLAY AT(12,15):USING
 "####":HC
2320 DISPLAY AT(12,21):USING
 "####":IC
2330 GOSUB 2690
2340 DISPLAY AT(24,2):"TASTE
 DRUECKEN FUER UMKEHR."
2350 CALL KEY(0,K,S):: IF S=
0 THEN 2350 ELSE 2380
2360 GOTO 2350
2370 ! SPIEGELUNG
2380 GOSUB 2040
```

```
2390 DISPLAY AT(3,8):"g..1..
k..2..k..3..k"
2400 DISPLAY AT(6,9):USING "
####":AC
2410 DISPLAY AT(6,15):USING
"####":DC
2420 DISPLAY AT(6,21):USING
"####":GC
2430 DISPLAY AT(9,9):USING "
####":BC
2440 DISPLAY AT(9,15):USING
"####":EC
2450 DISPLAY AT(9,21):USING
"####":HC
2460 DISPLAY AT(12,9):USING
"####":CB
2470 DISPLAY AT(12,15):USING
 "####":FC
2480 DISPLAY AT(12,21):USING
 "####":IC
2490 GOSUB 2690
2500 DISPLAY AT(15,7):"UMKEH
RMATRIX !"
2510 DISPLAY AT(17,9):"DETER
MINANTE :"
2520 DISPLAY AT(19,10):DET
2530 DISPLAY AT(24,2):"<< TA
STE DRUECKEN >>"
2540 CALL KEY(0,K,S):: IF S=
0 THEN 2540 ELSE 1100
2550 GOTO 2540
2560 DISPLAY AT(21,2):"DIE D
ETERMINANTE IST UNGUEL-TIG,
DA 0=0."
2570 GOTO 2230
2580 ! ENDE
2590 CALL CLEAR
2600 DISPLAY AT(9,5):"BIS ZU
M NAECHSTEN MAL."
2610 END
2620 DISPLAY AT(24,2):"KORRI
GIEREN <J> <N>"
2630 CALL KEY(0,K,S):: IF S=
0 THEN 2630
2640 IF K=78 OR K=110 THEN 2
680
2650 IF K=74 OR K=106 THEN 2
660 ELSE 2670
2660 IF XYZ=1 THEN 1220 ELSE
 IF XYZ=2 THEN 1400
2670 GOTO 2630
2680 RETURN
2690 FOR I=6 TO 12 STEP 3 ::
 CALL VCHAR(I,10,104):: CALL
 VCHAR(I,16,107):: CALL VCHA
R(I,22,107):: CALL VCHAR(I,2
8,107):: NEXT I
2700 RETURN
```

```
100 !
110 ! **********************
120 ! *                    *
130 ! *    >SOFT-PLOT<     *
140 ! *                    *
150 ! *    Copyright by    *
160 ! *                    *
170 ! *    Kay  Sievert    *
180 ! *                    *
190 ! *                    *
200 ! * Benoetigte Geraete *
210 ! *  TI99/4A  Konsole  *
220 ! *     Ext. Basic     *
230 ! *    Joystick (1)    *
260 ! *                    *
270 ! *  Speicherbelegung  *
280 ! *    11264 Bytes     *
290 ! *                    *
300 GOTO 370 :: CALL CLEAR :
: CALL SCREEN :: CALL KEY ::
 CALL JOYST :: CALL LOCATE :
: CALL MAGNIFY :: CALL SPRIT
E :: CALL SOUND :: CALL COLO
R :: CALL HCHAR :: CALL VCHA
R :: CALL GCHAR :: CALL DELS
PRITE
310 CALL CHARPAT :: CALL CHA
RSET :: CALL PATTERN
320 DIM Z$(10),CH$(100)
330 A :: B :: C :: D :: E ::
 F :: H :: I :: K :: L :: M
:: N :: O :: Q :: S :: U ::
V :: W :: X :: Y :: PR :: PR
S :: NR :: NC :: CN :: ROW :
: COL :: RA :: CA
340 CX :: CY :: CXM :: CYM :
: D1X :: D2X :: D1Y :: D2Y :
: XS :: YS :: XIS :: YIS ::
MA :: MB :: RAD :: ST :: PO
350 X$ :: Y$ :: C$
360 !@P-
370 CALL CLEAR :: CALL MAGNI
FY(2)
380 CALL SPRITE(#1,83,5,49,7
0,#2,79,5,49,86,#3,70,5,49,1
02,#4,84,5,49,116)
390 CALL SPRITE(#5,80,5,70,1
00,#6,76,5,70,116,#7,79,5,70
,132,#8,84,5,70,148)
400 X$="* (C)1985 by  KAY SI
EVERT *"
410 FOR I=1 TO 27 :: CALL HC
HAR(20,2+I,ASC(SEG$(X$,I,1))
):: CALL SOUND(-25,880,9)::
NEXT I
420 DISPLAY AT(24,4):"ERKLAE
RUNG ?(J/N):N" :: ACCEPT AT(
24,22)VALIDATE("JN")SIZE(-1)
:X$
430 DISPLAY AT(24,2):"[ALPHA
 LOCK]-TASTE LOESEN !" :: CA
```

```
LL SOUND(1000,110,10):: FOR
I=1 TO 500 :: NEXT I
440 IF X$="J" THEN A=1 :: GO
TO 450
450 CALL DELSPRITE(ALL):: CA
LL MAGNIFY(1):: CALL CLEAR
460 FOR I=0 TO 9 :: CALL CHA
RPAT(48+I,X$):: Z$(I)=X$ ::
NEXT I
470 C$="00000001001000110100
0101011001111000100110101011
1100110111101111"
480 W,F=1 :: PR,PRS=4 :: NR=
44 :: PO=2.5
490 !***** BILDSCHIRM ******
500 CALL SCREEN(1):: CALL HC
HAR(1,1,31,768)
510 !@P+
520 CALL CHAR(32,"0000000000
0000FF",33,"0101010101010101
",34,"FF",35,"80808080808080
80")
530 CALL CHAR(36,"02070E1C38
70608",37,"040E1F3E7CF8702",
38,"FF818181818181FF",39,"FF
C7BBBFBFBFBBC7")
540 CALL CHAR(40,"3C42818181
814238",41,"AA55AA55AA55AA55
",42,"F0F0F0F0")
550 !@P-
560 CALL HCHAR(1,3,32,28)::
CALL HCHAR(23,3,34,28):: CAL
L VCHAR(2,2,33,21):: CALL VC
HAR(2,27,33,21)
570 CALL VCHAR(2,31,35,21)::
 FOR I=5 TO 20 STEP 3 :: CAL
L HCHAR(I,28,34,3):: NEXT I
580 CALL HCHAR(13,28,30)
590 CALL HCHAR(3,29,37):: CA
LL HCHAR(6,29,38):: CALL HCH
AR(9,29,40):: CALL HCHAR(12,
29,34):: CALL HCHAR(15,29,39
)
600 CALL COLOR(0,6,1):: FOR
I=1 TO 14 :: CALL COLOR(I,4,
1):: NEXT I
610 IF A=1 THEN 1760
620 FOR I=45 TO 143 :: CALL
CHAR(I,"0"):: NEXT I :: GOSU
B 1640
630 ROW,COL=89 :: CALL SPRIT
E(#1,36,6,ROW-7,COL,#2,36,6,
200,100)
640 GOTO 1370
650 !**** VERGROESSERN *****
660 CALL GCHAR(INT(ROW/8),IN
T(COL/8)+1,V):: IF V=31 THEN
 1370
670 CALL CHARPAT(V,X$)
680 CALL CHAR(41,X$)
690 CALL SPRITE(#3,41,15,157
,221):: CALL MAGNIFY(2)
700 CALL KEY(1,Q,S):: IF Q<>
18 OR S=-1 THEN 700 ELSE CAL
L MAGNIFY(1):: CALL SPRITE(#
1,38,6,ROW-7,COL):: GOTO 137
0
710 !**** RADIEREN ******
720 U=ROW/8 :: V=COL/8
730 IF U-INT(U)>.5 THEN X=1
ELSE X=2
740 IF V-INT(V)>.5 THEN Y=1
ELSE Y=2
750 CALL GCHAR(U,V,Q):: IF Q
=31 THEN 1370 ELSE CALL CHAR
PAT(Q,X$)
760 IF X=1 AND Y=1 THEN X$="
0"&SEG$(X$,2,1)&"0"&SEG$(X$,
4,1)&"0"&SEG$(X$,6,1)&"0"&SE
G$(X$,8,9):: GOTO 800
770 IF X=1 AND Y=2 THEN X$=S
EG$(X$,1,1)&"0"&SEG$(X$,3,1)
&"0"&SEG$(X$,5,1)&"0"&SEG$(X
$,7,1)&"0"&SEG$(X$,9,8):: GO
TO 800
780 IF X=2 AND Y=1 THEN X$=S
EG$(X$,1,8)&"0"&SEG$(X$,10,1
)&"0"&SEG$(X$,12,1)&"0"&SEG$
(X$,14,1)&"0"&SEG$(X$,16,1):
: GOTO 800
790 X$=SEG$(X$,1,9)&"0"&SEG$
(X$,11,1)&"0"&SEG$(X$,13,1)&
"0"&SEG$(X$,15,1)&"0"
800 CXM,CYM=0 :: CALL CHAR(Q
,X$):: GOTO 1370
810 !***** LINIE ******
820 IF W=1 THEN A=ROW :: B=C
OL :: W=2 :: GOTO 1370 ELSE
IF W=2 THEN W=1 :: CALL LOCA
TE(#2,ROW-7,COL):: C=ROW ::
D=COL :: GOTO 970
830 !***** KREIS *******
840 IF W=1 THEN A=ROW :: B=C
OL :: W=2 :: GOTO 1370 ELSE
IF W=2 THEN W=1 :: CALL LOCA
TE(#2,ROW-7,COL):: C=ROW ::
D=COL :: GOTO 850
850 CALL LOCATE(#2,200,100):
: RAD=ABS(A-C):: U=A :: V=B
860 IF RAD>20 THEN D=1 ELSE
D=3
870 FOR I=1 TO 360 STEP D
880 A=U+INT(SIN(I*PI/180)*RA
D)
890 B=V+INT(COS(I*PI/180)*RA
D)
900 IF MA=A AND MB=B THEN 92
0 ELSE MA=A :: MB=B
910 GOSUB 1100
920 NEXT I :: ROW=U :: COL=V
 :: GOTO 1370
```

```
930 !**** CHARPLATZ ******
940 IF F=1 THEN F=2 :: CALL
GCHAR(ROW/8,COL/8+1,H):: CAL
L HCHAR(15,29,H):: GOTO 1370
950 CALL GCHAR(ROW/8,COL/8+1
,A):: IF A=H THEN CALL HCHAR
(15,29,39):: F=1 :: GOTO 137
0 ELSE CALL HCHAR(ROW/8,COL/
8+1,H):: GOTO 1370

960 !*** CALL DRAW ***
970 U=D-B :: V=C-A
980 D1X=SGN(V):: D1Y=SGN(U)
990 D2X=0 :: D2Y=SGN(V)
1000 M=ABS(V):: N=ABS(U)
1010 IF M>N THEN 1040
1020 D2Y=0 :: D2X=SGN(U)
1030 M=MAX(ABS(U),ABS(V)):
N=MIN(ABS(U),ABS(V))
1040 S=INT(M/2):: FOR I=0 TO
 M
1050 GOSUB 1100
1060 S=S+N
1070 IF S<M THEN A=A+D2Y ::
B=B+D2X :: GOTO 1090
1080 S=S-M :: A=A+D1X :: B=B
+D1Y
1090 NEXT I :: CALL LOCATE(#
2,200,100):: GOTO 1370

1100 !**********************
      ***** ++PLOTTEN++ ****
      **********************

1110 XS=B/8 :: XIS=INT(XS)::
 X=(XS-XIS)*8 :: IF XS=XIS T
HEN X=8
1120 YS=A/8 :: YIS=INT(YS)::
 Y=((YS-YIS)*8)*2 :: IF YS=Y
IS THEN Y=16
1130 CX=YIS+1 :: IF YS=YIS T
HEN CX=CX-1
1140 CY=XIS+1 :: IF XS=XIS T
HEN CY=CY-1
1150 IF CY<3 OR CY>26 THEN R
ETURN
1160 IF (CX=CXM AND CY=CYM)=
-1 THEN 1210
1170 CALL GCHAR(CX,CY,E)
1180 IF E>31 AND E<36 THEN W
=1 :: RETURN
1190 IF E=31 THEN NR=NR+1 ::
 E=NR :: IF NR>143 THEN 1740
 ELSE GOSUB 1640 :: NC=NR ::
 CALL HCHAR(CX,CY,NC)ELSE NC
=E
1200 CALL CHARPAT(NC,Y$)
1210 CXM=CX :: CYM=CY
1220 IF X<5 THEN Y=Y-1
1230 Q=ASC(SEG$(Y$,Y,1)):: I
F Q>57 THEN Q=Q-55 ELSE Q=Q-
48
1240 X$=SEG$(C$,1+Q*4,4)
1250 IF X>4 THEN X=X-4
1260 ON X GOTO 1270,1290,129
0,1280
1270 X$="1"&SEG$(X$,2,3):: G
OTO 1300
1280 X$=SEG$(X$,1,3)&"1" ::
GOTO 1300
1290 X$=SEG$(X$,1,X-1)&"1"&S
EG$(X$,X+1,4-X)
1300 Q=.5 :: L=0 :: FOR O=4
TO 1 STEP -1 :: IF SEG$(X$,O
,1)="1" THEN L=L+Q*2

1310 Q=Q*2 :: NEXT O
1320 IF L>9 THEN X$=CHR$(55+
L)ELSE X$=CHR$(L+48)
1330 Y$=SEG$(Y$,1,Y-1)&X$&SE
G$(Y$,Y+1,16-Y)
1340 CALL CHAR(NC,Y$)
1350 RETURN
1360 !**********************
      ***** GRIFFEL ******
1370 CALL KEY(0,Q,S):: IF S<
>0 THEN CALL SOUND(-100,440,
10):: CALL COLOR(0,6,1):: GO
TO 1440 ELSE CALL COLOR(0,3.
5+PO,1):: PO=-PO
1380 ST=PRS :: CALL JOYST(1,
X,Y):: CALL KEY(1,Q,S):: IF
Q<>18 THEN 1400
1390 IF X<>0 OR Y<>0 THEN ST
=.5 ELSE CALL SOUND(-100,880
,10):: ON PR GOTO 720,660,84
0,820,940
1400 IF Y=4 AND ROW>20 THEN
ROW=ROW-Y/ST ELSE IF Y=-4 AN
D ROW<162 THEN ROW=ROW-Y/ST
1410 IF X=-4 AND COL>28 THEN
 COL=COL+X/ST ELSE IF X=4 AN
D COL<208 THEN COL=COL+X/ST
1420 CALL LOCATE(#W,(ROW-7)+
RA,COL+CA):: GOTO 1370
1430 !**********************
      **** obtionen ******
1440 RA,CA=0
1450 CALL KEY(0,K,S):: IF S=
0 OR S=-1 THEN CALL COLOR(0,
3.5+PO,1):: PO=-PO :: GOTO 1
450
1460 X=0 :: IF K=101 THEN Y=
4 :: GOTO 1400 ELSE IF K=120
 THEN Y=-4 :: GOTO 1400 ELSE
 Y=0 :: IF K=115 THEN X=-4 :
: GOTO 1400 ELSE IF K=100 TH
EN X=4 :: GOTO 1400
1470 IF K=13 THEN CALL SOUND
(-100,131,10,196,15):: GOTO
1510
1480 IF K<>32 THEN 1370 ELSE
 CALL HCHAR(PR*3+1,28,31)::
CALL LOCATE(#2,200,100):: CA
LL HCHAR(15,29,39):: W,F=1
```

```
1490 IF K=32 THEN PR=PR+1 ::
 IF PR>7 THEN PR=1
1500 CALL HCHAR(PR*3+1,28,30
):: GOTO 1440
1510 ON PR GOTO 1520,1530,15
40,1550,1560,1570,1600
1520 GOSUB 1610 :: ROW=ROW+4
 :: COL=COL+4 :: CALL SPRITE
(#1,42,8,ROW,COL):: RA=3 ::
CA=-4 :: PRS=1 :: GOTO 1370
1530 GOSUB 1610 :: CALL PATT
ERN(#1,38):: RA=-1 :: PRS=.5
 :: GOTO 1370

1540 CALL SPRITE(#1,36,6,ROW
-7,COL):: PRS=4 :: GOTO 1370
1550 CALL SPRITE(#1,36,6,ROW
-7,COL):: PRS=4 :: GOTO 1370
1560 GOSUB 1610 :: CALL SPRI
TE(#1,38,5,ROW,COL):: RA=-1
:: PRS=.5 :: GOTO 1370
1570 FOR A=2 TO 22 :: FOR B=
3 TO 26 :: CALL GCHAR(A,B,C)
:: IF C=31 THEN 1590 ELSE IF
 C=NR+1 THEN A=22 :: B=26 ::
 GOTO 1590
1580 CALL CHARPAT(C,X$):: CH
$(C-44)=X$&"/"&STR$(A)&"-"&S
TR$(B)
1590 NEXT B :: NEXT A :: GOT
O 1700

1600 FOR A=2 TO 22 :: CALL H
CHAR(A,3,31,24):: NEXT A ::
FOR A=45 TO 143 :: CALL CHAR
(A,"0"):: NEXT A :: NR=44 ::
 GOSUB 1640 :: GOTO 1440
1610 ROW=INT(ROW/8)*8+1 :: C
OL=INT(COL/8)*8+1 :: RETURN
1620 !***** CHAR ANZEIGE ***
1630 !*** DISPLAY *****
1640 IF LEN(STR$(143-NR))=2
THEN 1660
1650 CALL CHAR(43,Z$(143-NR)
):: CN=31 :: GOTO 1680
1660 X$=STR$(143-NR):: CALL
CHAR(44,Z$(ASC(SEG$(X$,1,1))
-48))
1670 CN=44 :: CALL CHAR(43,Z
$(ASC(SEG$(X$,2,1))-48))
1680 CALL HCHAR(18,30,43)::
CALL HCHAR(18,29,CN):: RETUR
N
1690 !**** AUSDRUCKEN ****
1700 CALL CHARSET :: CALL DE
LSPRITE(ALL):: CALL CLEAR ::
 CALL SCREEN(8)
1710 FOR I=1 TO 99 :: PRINT
I+44;CH$(I): :
1720 CALL KEY(0,K,S):: IF S=
0 THEN 1720
1730 NEXT I :: GOTO 1710
1740 F,W=1 :: NR=NR-1 :: CAL
L LOCATE(#2,200,100):: GOTO
1370
1750 !**** ERKLAERUNG ****
1760 FOR I=65 TO 90 :: CALL
CHARPAT(I,X$):: CALL CHAR(I+
32,X$):: NEXT I :: FOR I=9 T
O 12 :: CALL COLOR(I,2,4)::
NEXT I
1770 CALL CHAR(123,"00000402
FF0204",125,"00000000FC",126
,"0")
1780 CALL SPRITE(#1,123,16,2
00,205):: F=7.5
1790 READ E,B :: IF E=1000 T
HEN 1890
1800 FOR A=1 TO B :: READ C,
X$
1810 FOR I=1 TO LEN(X$):: IF
 SEG$(X$,I,1)=" " THEN D=31
ELSE D=ASC(SEG$(X$,I,1))
1820 CALL HCHAR(C,3+I,D):: N
EXT I :: NEXT A :: CALL LOCA
TE(#1,E,205)
1830 DISPLAY AT(24,4)SIZE(20
):RPT$(CHR$(31),20)
1840 CALL KEY(0,K,S):: IF S<
>0 THEN GOSUB 1880 :: GOTO 1
790
1850 CALL COLOR(#1,8.5+F)::
F=-F
1860 FOR I=1 TO 50 :: NEXT I
 :: DISPLAY AT(24,4)SIZE(20)
:"bitte";"~";"taste";"~";"dr
uecken" :: CALL COLOR(#1,8.5
+F):: F=-F
1870 FOR I=1 TO 100 :: NEXT
I :: GOTO 1830
1880 FOR I=2 TO 22 :: CALL H
CHAR(I,4,31,23):: NEXT I ::
RETURN
1890 F=1 :: CALL DELSPRITE(#
1):: FOR I=9 TO 12 :: CALL C
OLOR(I,4,1):: NEXT I :: GOTO
 620
1900 !@P+
1910 DATA 200,10,4,".....so
ft}plot",6,"soft}plot DIENT
ZUR",8,"EINFACHEN ERSTELLUNG
",10,"EINFARBIGER GRAPHIKEN.
"
1920 DATA 12,"ES KOENNEN INS
GESAMMT",14,"99 ZEICHEN =633
4 DOTS",16,"GESETZT WERDEN.
DAMIT",18,"SIE soft}plot SIN
NVOLL"
1930 DATA 20,"ANWENDEN,BEACH
TEN SIE",22,"BITTE FOLGENDE
PUNKTE:"
1940 DATA 200,10,3,"DURCH DR
UECKEN VON",5,"space KOENNEN
```

```
 SIE",7,"ZWISCHEN DEN VERSCH
IE-",9,"DENEN OBTIONEN WAEHL
EN.",11,"DRUECKEN VON enter"
1950 DATA 13,"FIXIERT DIE GE
WUENSCHTE",15,"FUNKTION.DER
BLINKENDE",17,"CURSOR ZEIGT
IHNEN,",19,"WELCHE OPTION GE
RADE"
1960 DATA 21,"AUSGEFUEHRT WI
RD."
1970 DATA 200,10,3,"joystick
fkt:",5,"PUNKTWEISE BEWEGUNG
 DES",7,"ZEICHENSTIFTES. UM
DEN",9,"STIFT SCHNELLER ZU B
E-",11,"WEGEN,JOYSTICK IN DI
E"

1980 DATA 13,"GEWUENSCHTE RI
CHTUNG ",15,"DANN button [=
SCHUSS-",17,"KNOPF] DRUECKEN
.",19,"pfeiltasten: PUNKTWEI
SE",21,"BEWEGUNG DES STIFTES
."
1990 DATA 17,10,3,"radieren:
",5,"DURCH DRUECKEN DES",7,"
button WIRD DAS 4/4",9,"DOT
GROSSE QUADRAT",11,"UNTER DE
M RADIERGUMMI"
2000 DATA 13,"GELOESCHT UND
KANN",15,"NUN NEU GEPLOTTET"
,17,"WERDEN.",19,"RUECKSPRUN
G ZUM MENUE",21,"DURCH space
."
2010 DATA 41,10,3,"vergroess
ern:",5,"DAS UNTER DEM QUADR
AT",7,"LIEGENDE ZEICHEN WIRD
",9,"NACH DRUECKEN DES BU-",
11,"TON IN UNTERSTEN"
2020 DATA 13,"FELD VERGROESS
ERT",15,"DARGESTELLT. UM DAS
",17,"QUADRAT WEITERBEWEGEN"
,19,"ZU KOENNEN,DEN BUTTON",
21,"NOCHMAL DRUECKEN."
2030 DATA 65,10,3,"kreis)fun
ktion:",5,"BUTTON DRUECKEN,U
M DEN",7,"MITTELPUNKT DES KR
EISES",9,"FESTZULEGEN.STIFT
NACH",11,"OBEN/UNTEN BEWEGEN
,UM"
2040 DATA 13,"DEN RADIUS FES
TZULEGEN.",15,"DANN NOCHMAL
DEN BUTTON",17,"DRUECKEN,UND
 DAS PRO-",19,"GRAMM PLOTTET
 DEN",21,"GEWUENSCHTEN KREIS
."
2050 DATA 89,10,3,"linie)fun
ktion:",5,"BUTTON DRUECKEN,U
M DEN",7,"ANF.PUNKT FESTZULE
GEN,",9,"DANN STIFT 2 ZUM EN
D-",11,"PUNKT FUEHREN UND DE
N"
2060 DATA 13,"BUTTON DRUECKE
N.",15,"laengeres HALTEN DES
",17,"BUTTON PLOTTET EINEN",
19,"PUNKT. DIES WIRD DURCH",
21,"2 TOENE ANGEZEIGT."
2070 DATA 113,19,3,"kopier)f
unktion:",5,"DIENT ZUM SCHNE
LLEN",6,"SETZEN VON GLEICHAR
TI-",7,"GEN ZEICHEN.DAS QUAD
RAT",8,"UEBER DAS ZEICHEN FU
E-"
2080 DATA 9,"HREN UND DURCH
BUTTON",10,"SPEICHERN.DRUECK
EN DES",11,"BUTTON KOPIERT E
S NUN",12,"AN JEDER GEWUENSC
HTEN",13,"STELLE. zeichentas
ch:"

2090 DATA 14,"ZEICHEN SETZEN
 - DANN",15,"WIEDER BUTTON.N
UN KANN",16,"EIN NEUES ZEICH
EN",17,"AUFGENOMMEN WERDEN."
,18,"vorsicht:WIRD EIN MEHR-
"

2100 DATA 19,"FACH KOPIERTES
 ZEICHEN",20,"NACHGEPLOTTET,
AENDERN",21,"SICH DAMIT AUCH
 ALLE",22,"KOPIEN DES ZEICHE
NS."
2110 DATA 137,10,3,"zeichen)
ausdruck:",5,"UM EINE GRAPHI
K AUCH",7,"IN ANDEREN PROGRA
MMEN",9,"VERWENDEN ZU KOENNE
N,",11,"KANN MAN SICH DEN"
2120 DATA 13,"CHARACTERCODE
UND DIE",15,"POSITION DER ZE
ICHEN",17,"MIT space AUFLIST
EN .",19,"DAMIT IST DAS PROG
RAMM",21,"JEDOCH beendet."
2130 DATA 161,7,3,"loesch)fu
nktion:",5,"MIT DIESER OBTIO
N",7,"KOENNEN SIE DEN BILD-"
,9,"SCHIRM LOESCHEN. DER"
2140 DATA 11,"ZEICHENVORRAT
STEHT",13,"DANNACH WIEDER vo
ll",15,"ZUR VERFUEGUNG."
2150 DATA 200,9,5," -ZUM SCH
LUSS EIN TIP-",7,"MACHEN SIE
 SICH SCHRITT",9,"FUER SCHRI
TT MIT DEN",11,"FUNKTIONEN D
ES PRO-",13,"GRAMMS VERTRAUT
,DENN"
2160 DATA 15,"NUR SO WERDEN
SIE DEN",17,"OPTIMALEN NUTZE
N AUS",19,"DEM PROGRAMM ZIEH
EN.",22,".....viel-spass"
2170 DATA 1000,1000
```

ACHTUNG!
089/1298011 Ab 1.10.85!

MATHE TABELLEN TORE

Tabellenberechnung

Nach Erscheinen des Titelbilds wird nach dem Tagesdatum sowie der Benutzung eines Drukkers gefragt. Hier besteht die Möglichkeit, bei seriellen Druckern die Bezeichnung RS232 ... (weitere Bez. je nach Typ), oder bei parallelen Druckern die Bezeichnung PI0 ... (weitere Bez. je nach Typ) anzugeben.

Im Menue sind folgende Optionen vorgesehen: — Arbeitsblatt — Daten einlesen — Hardcopy — Daten speichern — Berechnung nach Formeln — Help-Funktion

Hier wird auch die Dateibezeichnung sowie das bei Eröffnung eingegebene Tagesdatum angezeigt.

Arbeitsblatt: Es können max. 10 Textbezeichnungen sowie 10 Werte erfaßt werden. Sollte ein Wert nicht unmittelbar unter den nächsten geschrieben werden, so ist in dieser Wert-Spalte, die übersprungen werden soll, eine „0" (Null) einzugeben. Wenn dies nicht geschieht, werden die nachfolgenden Werte nicht bearbeitet bzw. berücksichtigt.

Bei Text ist dies unerheblich.

Am unteren Bildschirmende wird die Gesamtsumme ausgeworfen. Außerdem wird während des Dialogs permanent die Möglichkeit der weiteren Vorgehensweise angezeigt.

Es können jederzeit Änderungen im Text oder im Wertbereich vorgenommen werden. Nach Eingabe bzw. Änderung können Texte und Werte (der Übersichtlichkeit halber) mit der Funktion „T" tabellarisch aufgelistet werden.

Mit Funktion „P" werden die eingegebenen Werte prozentual pro Position ausgewertet. Diese können wiederum mit der Funktion „G" graphisch dargestellt werden.

Hierbei werden allerdings nur die positiven Werte angezeigt (+ Beträge). Durch nochmaliges Drücken der Taste „t" wird wieder die Tabelle aufgelistet. Funktion „C" steht für Hardcopy, die ebenfalls im Arbeitsblatt möglich ist. Graphikausdruck ist allerdings auf Grund der begrenzten Speicherkapazität nicht vorgesehen.

Die Funktion „F" ermöglicht die Berechnung nach Formeln (bereits integriert).

Durch Drücken der Taste „M" wird der Mittelwert der eingegebenen Werte ausgeworfen.

Daten einlesen: Es können mit diesem Programm bereits erstellte Daten eingelesen und bearbeitet werden. Der bei der Eröffnung angegebene Dateiname sowie das Tagesdatum wird mit dem Einspielen der neuen Datei automatisch überschrieben und auf neuen Stand gebracht. Nach Beendigung des Ladevorgangs verzweigt das Programm zum Menue und die neu geladene Datei sowie das Erstellungsdatum wird angezeigt.

Sollten beim Ladevorgang Fehler auftreten (File oder Data-Error) so wird dies durch eine Mitteilung angezeigt. Soll trotzdem eine neue Datei eingelesen werden, so ist das Programm abzubrechen und wieder neu zu starten.

Daten abspeichern: Es werden nur die im Arbeitsblatt erstellten Werte und Texte gesichert, nicht aber die prozentualen oder graphischen Auswertungen.

Help-Funktion: Mit dieser Funktion kann jederzeit, im Menue oder im Arbeitsblatt, festgestellt werden, welche Bedeutung die angegebenen Parameter haben.

Gerhard Hübner

Torschießen

Torschießen ist ein Spiel für zwei Spieler. Der eine Spieler schießt und der andere wehrt ab. Für jedes Tor oder gehaltenen Ball gibt es einen Punkt. Wer zuerst 10 Punkte hat, hat gewonnen.

Steuerung: siehe Programm.
Achtung:

1. Berühren sich die Spieler, so wird „FOUL" angezeigt und der Angriff wiederholt.
2. Kommt der Torwart zu weit aus seinem Tor, kann er sich nur ganz langsam bewegen.

Programmerläuterungen:
Hauptprogramm 220—400
Sub Punkte 410—470 (Spieler)
Sub Gepunk 480—540 (Torwart)
Sub Gehalten 550—610
Sub Stand 620—649 (Spielstand)
Sub Schuß 650—750
Sub Foul 760—800
Sub Bildschirm 810—1080 (Graphik)
Sub Spielanleitung 1090—1300
Sub Torwart 1310—1360 (Torwartsteuerung)
Sprite 1 Spieler
Sprite 2 Torwart
Sprite 3 Ball
Empfindlichkeit des Torwarts ist in den „Call Coinc(...)"-Anweisungen festgelegt (Zeile 280,340, 600).

```
100 ! ************************
110 ! *                      *
120 ! *   TORESCHIESSEN      *
130 ! *                      *
140 ! *   Copyright  by      *
150 ! *                      *
160 ! *   Steffen Trefz      *
170 ! *                      *
180 ! *                      *
190 ! * Benoetigte Geraete   *
200 ! *  TI99/4A  Konsole    *
210 ! *     Ext. Basic       *
220 ! *   Joystick (1&2)     *
230 ! *                      *
260 ! *  Speicherbelegung    *
270 ! *     5610 Bytes       *
280 ! *                      *
290 ! ************************
300 CALL CLEAR :: CALL SPIEL
ANLEITUNG :: CALL BILDSCHIRM
310 CALL SPRITE(#1,124,11,16
5,90):: CALL SPRITE(#2,112,1
6,165,125):: CALL SPRITE(#3,
136,8,96,130)
320 CALL JOYST(1,X,Y):: CALL
 TORWART :: CALL KEY(0,K,S):
: IF K=32 THEN CALL DELSPRIT
E(ALL):: CALL CLEAR :: END
330 CALL MOTION(#1,-Y*2,X*2)
340 CALL COINC(#1,#2,6,T)::
CALL COINC(#2,#3,10,P):: CAL
L COINC(#1,#3,6,Q)
350 IF Q=-1 THEN CALL FOUL :
: GOTO 310 ELSE 360
360 IF P=-1 THEN CALL GEPUNK
(G,W):: CALL GEHALTEN(G,W)::
 GOTO 460 ELSE 370
370 IF T=-1 THEN CALL SCHUSS
 :: GOTO 320 ELSE 380
380 CALL POSITION(#2,V,H)
390 CALL JOYST(1,X,Y):: CALL
 MOTION(#1,-Y*2,X*2):: CALL
TORWART
400 CALL COINC(#2,#3,10,U)::
 CALL COINC(#1,#3,6,F)
410 IF F=-1 THEN CALL FOUL :
: GOTO 310 ELSE IF U=-1 THEN
 CALL GEPUNK(G,W):: CALL GEH
ALTEN(G,W):: GOTO 460 ELSE 4
20
420 CALL POSITION(#2,V,H)
430 IF U=0 THEN IF V<90 AND
V>20 THEN IF H>89 AND H<157
THEN CALL PUNKTE(G,W):: GOTO
 450 ELSE 440
440 IF V<89 OR H>250 OR H<10
 THEN CALL DELSPRITE(#2):: C
ALL HCHAR(2,12,40,10):: DISP
LAY AT(2,12)SIZE(6):"VORBEI"
 :: GOTO 450 ELSE 320
450 CALL MOTION(#1,0,0,#2,0,
0,#3,0,0):: CALL STAND(G,W):
: FOR I=1 TO 250 :: NEXT I :
: GOTO 310
460 CALL POSITION(#2,R,S)::
IF R>=180 THEN 310 ELSE CALL
 STAND(G,W):: GOTO 320
470 SUB PUNKTE(G,W)
480 CALL SOUND(100,523,2,659
,2):: CALL HCHAR(2,12,40,10)
490 G=G+1 :: DISPLAY AT(2,12
)SIZE(3):"TOR" :: IF G=10 TH
EN DISPLAY AT(2,12)SIZE(4):"
ENDE" :: GOTO 500 ELSE CALL
STAND(G,W):: SUBEXIT
500 CALL KEY(0,K,S):: IF K=3
2 THEN CALL DELSPRITE(ALL)::
 CALL CLEAR :: END
510 CALL KEY(2,K,S):: CALL M
OTION(#1,0,0,#2,0,0,#3,0,0):
: IF K=18 THEN G,W=0 :: DISP
LAY AT(2,12)SIZE(5):"START"
:: SUBEXIT ELSE 500
520 SUBEND
530 SUB GEPUNK(G,W)
540 CALL SOUND(100,261,2,326
,2):: CALL HCHAR(2,12,40,10)
550 W=W+1 :: IF W=10 THEN DI
SPLAY AT(2,12)SIZE(4):"ENDE"
 :: GOTO 560 ELSE CALL STAND
(G,W):: SUBEXIT
560 CALL KEY(0,K,S):: IF K=3
2 THEN CALL DELSPRITE(ALL)::
 CALL CLEAR :: END
570 CALL KEY(1,K,S):: CALL M
OTION(#1,0,0,#2,0,0,#3,0,0):
: IF K=18 THEN G,W=0 :: DISP
LAY AT(2,12)SIZE(5):"START"
:: SUBEXIT ELSE 560
580 SUBEND
590 SUB GEHALTEN(G,W)
600 A=16 :: B=3 :: CALL PATT
ERN(#1,124):: CALL MOTION(#2
,A,B,#3,0,0,#1,0,0)
610 DISPLAY AT(2,12)SIZE(8):
"GEHALTEN" :: CALL STAND(G,W
)
620 FOR I=1 TO 200 :: NEXT I
 :: CALL MOTION(#2,0,0)
630 CALL POSITION(#2,V,H)::
IF V>=180 THEN SUBEXIT ELSE
640
640 CALL COINC(#2,#3,10,U)::
 IF U=-1 THEN 600 ELSE SUBEX
IT
650 SUBEND
660 SUB STAND(G,W)
670 DISPLAY AT(4,12)SIZE(1):
CHR$(G+48):: DISPLAY AT(4,15
)SIZE(1):":" :: DISPLAY AT(4
,18)SIZE(1):CHR$(W+48)
```

```
680 SUBEND
690 SUB SCHUSS
700 CALL KEY(1,K,S):: CALL J
OYST(1,X,Y):: CALL TORWART !
:: CALL JOYST(2,X1,Y1)
710 CALL MOTION(#1,-Y*2,X*2)
:: CALL MOTION(#2,-Y*2,X*2)
720 CALL COINC(#1,#3,6,Q)::
IF Q=-1 THEN CALL FOUL :: SU
BEXIT ELSE 730
730 IF K=18 AND Y=0 AND X=4
THEN N=-20 :: L=15 :: GOTO 7
80 :: SUBEXIT ELSE 740
740 IF K=18 AND Y=0 AND X=-4
 THEN N=-20 :: L=-15 :: GOTO
 780 :: SUBEXIT ELSE 750
750 IF K=18 AND Y=4 AND X=4
THEN N=-20 :: L=8 :: GOTO 78
0 :: SUBEXIT ELSE 760
760 IF K=18 AND Y=4 AND X=-4
 THEN N=-20 :: L=-8 :: GOTO
780 :: SUBEXIT ELSE 770
770 IF K=18 AND Y=4 AND X=0
THEN N=-20 :: L=0 :: GOTO 78
0 :: SUBEXIT ELSE 700
780 CALL PATTERN(#1,120):: C
ALL MOTION(#2,N,L)
790 SUBEND
800 SUB FOUL
810 CALL COINC(#1,#3,6,Q)::
IF Q=-1 THEN CALL HCHAR(2,12
,40,10):: DISPLAY AT(2,12)SI
ZE(4):"FOUL" :: GOTO 820
820 CALL DELSPRITE(#3):: CAL
L MOTION(#2,0,0,#1,0,0):: CA
LL PATTERN(#1,140)
830 FOR I=1 TO 220 :: NEXT I
 :: CALL DELSPRITE(ALL):: SU
BEXIT
840 SUBEND
850 SUB BILDSCHIRM
860 CALL SCREEN(2):: CALL CH
AR(33,"3C3C99A5A55A3C3C",40,
RPT$("F",16),104,RPT$("F",16
))
870 CALL CHAR(96,"0707070707
0707FF",97,RPT$("07",9),98,R
PT$("E0",8),99,"E0E0E0E0E0E0
E0FF",100,"FFFFFFFF")
880 CALL CHAR(116,"000000000
00000FF",102,"FFFFFFFFE0E0E0
E0",103,"FFFFFFFF07070707")
890 CALL CHAR(120,"030303010
3070F1B3303030303030307C0C0C
080C0E0F0DCC0C0E07030303800"
)
900 CALL CHAR(124,"030303011
F1F1B1B0303033F3F20000080808
000C0C0F8F880F0F0303030380")
910 CALL CHAR(112,RPT$("0",5
2)&"183C3C18")
920 CALL CHAR(136,"101313131
11F0F0707070706060606E10909
0901C0F0EC0C0C0C0C0C0C0C0C0E")
930 CALL CHAR(140,"38BABA93F
F3938397B7F1F3F266C181000000
0C0C0F8A0C080E0B09080C")
940 CALL HCHAR(8,15,100,7)::
 CALL VCHAR(9,14,98,3):: CAL
L VCHAR(9,21,97,3)
950 CALL HCHAR(8,14,102):: C
ALL HCHAR(8,21,103)
960 CALL VCHAR(12,14,99):: C
ALL VCHAR(12,21,96)
970 CALL HCHAR(12,1,116,13):
: CALL HCHAR(12,15,116,6)::
CALL HCHAR(12,22,116,11)
980 CALL HCHAR(1,1,33,43)
990 FOR I=12 TO 23 :: FOR J=
2 TO 5 :: CALL HCHAR(J,I,40)
:: NEXT J :: NEXT I
1000 FOR I=2 TO 5 :: CALL HC
HAR(I,24,33,20):: NEXT I
1010 CALL HCHAR(6,12,33,54)
1020 FOR I=8 TO 11 :: FOR J=
1 TO 13 :: CALL HCHAR(I,J,10
4):: NEXT J :: NEXT I
1030 FOR I=8 TO 10 :: FOR J=
22 TO 32 :: CALL HCHAR(I,J,1
04):: NEXT J :: NEXT I
1040 FOR J=1 TO 13 :: CALL H
CHAR(11,J,104):: FOR K=22 TO
 32 :: CALL HCHAR(11,K,104):
: NEXT K :: NEXT J
1050 FOR I=9 TO 11 :: FOR J=
15 TO 20 :: CALL HCHAR(I,J,1
04):: NEXT J :: NEXT I
1060 ! COLORS **************
1070 CALL COLOR(2,11,11,8,16
,1,9,16,10,10,10,10,11,16,10
)
1080 FOR I=3 TO 7 :: CALL CO
LOR(I,2,11):: NEXT I
1090 CALL MAGNIFY(3):: CALL
SCREEN(13)
1100 SUBEND
1110 SUB SPIELANLEITUNG
1120 CALL CLEAR :: DISPLAY A
T(10,4):"WUENSCHEN SIE EINE
SPIELAN- LEITUNG ?(J/N)"
1130 CALL KEY(0,K,S):: IF S=
0 THEN 1130
1140 IF K=74 OR K=106 THEN 1
150 ELSE SUBEXIT
1150 CALL CLEAR :: CALL SCRE
EN(8)
1160 PRINT "******* TORSCHIE
SSEN *******" :: PRINT :;:;:
;
1170 PRINT "SPIELERANLEITUNG
:" :: PRINT "STEUERUNG ERFOL
```

```
GT UEBER.....JOYSTICK-1"
1180 PRINT "DER SCHUSS WIRD
DURCH KNOPF-DRUCK UND DRUECK
EN IN ENT-..SPRECHENDE RICHT
UNG AUSGE-..LOEST !"
1190 PRINT ;;;;;;: PRINT "BI
TTE EINE TASTE DRUECKEN"
1200 CALL KEY(0,K,S):: IF S=
0 THEN 1200
1210 CALL CLEAR
1220 PRINT "TORWARTANLEITUNG
:"
1230 PRINT "STEUERUNG ERFOLG
T UEBER.....JOYSTICK-2"
1240 PRINT "DER TORWART KANN
 DIE SCHUSS-RI ABWEHREN UND
SO PUNKTEN"
1250 PRINT ;;;;;;;;;: PRINT
 "BITTE EINE TASTE DRUECKEN"
1260 CALL KEY(0,K,S):: IF S=
0 THEN 1260
1270 CALL CLEAR
1280 PRINT " NEUER START AUF
 KNOPFDRUCK..DES VERLIERERS"
;;;;;;;: PRINT " ENDE AUF LE
ERTASTE";;;;;;;: PRINT " ALP
HALOCK HERAUSNEHMEN!";;;;;;;
;
1290 PRINT " BITTE EINE TAST
E DRUECKEN"
1300 CALL KEY(0,K,S):: IF S=
0 THEN 1300 ELSE CALL CLEAR
:: SUBEXIT
1310 SUBEND
1320 SUB TORWART
1330 CALL JOYST(2,X1,Y1)
1340 CALL MOTION(#3,-Y1*2,X1
*2)
1350 CALL POSITION(#3,R,S)
1360 IF R>=140 THEN CALL MOT
ION(#3,0,0)
1370 SUBEND
```


UNSER ☎ TELEFONSERVICE ☎

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite

ACHTUNG !!!

Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13


```
100 ! ***********************
110 ! *                     *
120 ! *TABELLENBERECHNUNGEN*
130 ! *                     *
140 ! *    Copyright by     *
150 ! *                     *
160 ! *  Peter Eisenbraun   *
170 ! *                     *
190 ! * Benoetigte Geraete  *
200 ! *  TI99/4A  Konsole   *
210 ! *     Ext. Basic      *
220 ! *                     *
260 ! *  Speicherbelegung   *
270 ! *    11184 Bytes      *
280 ! *                     *
290 ! ***********************
295 !
300 CALL SCREEN(2):: FOR T=9
 TO 12 :: CALL COLOR(T,16,5)
:: NEXT T :: FOR I=5 TO 8 ::
 CALL COLOR(I,8,2):: NEXT I
310 FOR U=0 TO 4 :: CALL COL
OR(U,11,2):: NEXT U :: DISPL
AY AT(5,3)ERASE ALL:"   TABE
LLENBERECHNUG   " :: FOR DL=
1 TO 500 :: NEXT DL
320 ON WARNING NEXT :: DISPL
AY AT(3,10)ERASE ALL:"TT.MM.
JJ" :: DISPLAY AT(9,10):"DAT
EINAME" :: ACCEPT AT(5,10)SI
ZE(2)VALIDATE(DIGIT):TT
330 ACCEPT AT(5,13)SIZE(2)VA
LIDATE(DIGIT):MM :: ACCEPT A
T(5,16)SIZE(2)VALIDATE(DIGIT
):JJ
340 IF TT<1 OR TT>31 OR MM<1
 OR MM>12 THEN 320 ELSE ACCE
PT AT(11,10)SIZE(10)VALIDATE
(UALPHA,"."):DN$
350 DISPLAY AT(15,10):"DRUCK
ER J N" :: ACCEPT VALIDATE("
JN"):EG$ :: IF EG$="J" THEN
DISPLAY AT(17,1):"BEZEICHNUN
G ?"
360 IF EG$="N" THEN 370 :: D
ISPLAY AT(19,1):"RS232.BA=..
.ODER PIO ?" :: ACCEPT AT(23
,1)VALIDATE(UALPHA,".=/-1234
567890"):DR$
370 DISPLAY AT(3,11)ERASE AL
L:"auswahl" :: DISPLAY AT(8,
1):"a>ARBEITSBLATT...r>READ
DATA" :: DISPLAY AT(10,1):"h
>HELP ROUTINE...s>SAVE DATA"
380 DISPLAY AT(12,1):"c>HARD
COPY" :: DISPLAY AT(22,1):"d
atum ";TT;".";MM;".";JJ :: D
ISPLAY AT(24,1):"datei ";TAB
(8);DN$
390 DISPLAY AT(18,1):"drucke
r";TAB(20);DR$ :: CALL KEY(0
```

```
,K,S):: IF S=0 THEN 390 ELSE
 400
400 IF K=65 THEN 410 :: IF K
=67 THEN 1580 :: IF K=83 THE
N 880 :: IF K=82 THEN 920 ::
 IF K=72 THEN 1540 ELSE 370
410 DISPLAY AT(1,1)ERASE ALL
:F1$ :: DISPLAY AT(3,1):F2$
:: DISPLAY AT(5,1):F3$
420 DISPLAY AT(1,16):USING "
######.##":Z1 :: DISPLAY AT(
3,16):USING "######.##":Z2 :
: DISPLAY AT(5,16):USING "##
####.##":Z3
430 DISPLAY AT(7,1):F4$ :: D
ISPLAY AT(9,1):F5$ :: DISPLA
Y AT(11,1):F6$ :: DISPLAY AT
(13,1):F7$
440 DISPLAY AT(7,16):USING "
######.##":Z4 :: DISPLAY AT(
9,16):USING "######.##":Z5 :
: DISPLAY AT(11,16):USING "#
#####.##":Z6
450 DISPLAY AT(13,16):USING
"######.##":Z7
460 DISPLAY AT(15,1):F8$ ::
DISPLAY AT(17,1):F9$ :: DISP
LAY AT(19,1):F10$
470 DISPLAY AT(15,16):USING
"######.##":Z8 :: DISPLAY AT
(17,16):USING "######.##":Z9
 :: DISPLAY AT(19,16):USING
"######.##":Z10
480 DISPLAY AT(1,27):"1" ::
DISPLAY AT(3,27):"2" :: DISP
LAY AT(5,27):"3" :: DISPLAY
AT(7,27):"4" :: DISPLAY AT(9
,27):"5" :: DISPLAY AT(11,27
):"6"
490 DISPLAY AT(13,27):"7" ::
 DISPLAY AT(15,27):"8" :: DI
SPLAY AT(17,27):"9" :: DISPL
AY AT(19,26):"10"
500 DISPLAY AT(22,1):"SUMME
TOTAL" :: ST=Z1+Z2+Z3+Z4+Z5+
Z6+Z7+Z8+Z9+Z10 :: DISPLAY A
T(22,16):USING "######.##":S
T
510 DISPLAY AT(24,3):"~~text
e~aend~einf~jin~" :: CALL KE
Y(0,K,S):: IF S=0 THEN 510 :
: IF K=74 THEN 540 :: IF K=7
8 THEN 520 ELSE 510
520 DISPLAY AT(24,3):"~~wert
e~aend~einf~jin~" :: CALL KE
Y(0,K,S):: IF S=0 THEN 520 :
: IF K=74 THEN 670 ELSE 530
530 IF K=78 THEN 780 ELSE 37
0
540 ON WARNING NEXT :: DISPL
AY AT(24,3):"~~welche~poitio
ns~nr~~" :: ACCEPT AT(24,26)
SIZE(2)BEEP VALIDATE(DIGIT):
PS :: GOTO 560
550 DISPLAY AT(24,3):"~~inse
rt~neuen~text~~~" :: RETURN
560 IF PS<1 OR PS>10 THEN 54
0 :: IF PS=1 THEN GOSUB 550
:: ACCEPT AT(1,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F1$ :
: GOTO 510
570 IF PS=2 THEN GOSUB 550 :
: ACCEPT AT(3,1)SIZE(10)BEEP
 VALIDATE(UALPHA,"."):F2$ ::
 GOTO 510
580 IF PS=3 THEN GOSUB 550 :
: ACCEPT AT(5,1)SIZE(10)BEEP
 VALIDATE(UALPHA,"."):F3$ ::
 GOTO 510
590 IF PS=4 THEN GOSUB 550 :
: ACCEPT AT(7,1)SIZE(10)BEEP
 VALIDATE(UALPHA,"."):F4$ ::
 GOTO 510
600 IF PS=5 THEN GOSUB 550 :
: ACCEPT AT(9,1)SIZE(10)BEEP
 VALIDATE(UALPHA,"."):F5$ ::
 GOTO 510
610 IF PS=6 THEN GOSUB 550 :
: ACCEPT AT(11,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F6$ :
: GOTO 510
620 IF PS=7 THEN GOSUB 550 :
: ACCEPT AT(13,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F7$ :
: GOTO 510
630 IF PS=8 THEN GOSUB 550 :
: ACCEPT AT(15,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F8$ :
: GOTO 510
640 IF PS=9 THEN GOSUB 550 :
: ACCEPT AT(17,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F9$ :
: GOTO 510
650 IF PS=10 THEN GOSUB 550
:: ACCEPT AT(19,1)SIZE(10)BE
EP VALIDATE(UALPHA,"."):F10$
 :: GOTO 510
660 DISPLAY AT(24,3)SIZE(20)
:"~~insert~neuen~wert~~~" ::
 RETURN
670 ON WARNING NEXT :: DISPL
AY AT(24,5)SIZE(20):"welche~
positions~nr~" :: ACCEPT AT(
24,26)SIZE(2)BEEP VALIDATE(D
IGIT):PP :: GOTO 680
680 IF PP<1 OR PP>10 THEN 67
0 :: IF PP=1 THEN GOSUB 660
:: ACCEPT AT(1,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z1 :: GO
TO 500
690 IF PP=2 THEN GOSUB 660 :
```

```
: ACCEPT AT(3,15)SIZE(9)BEEP
 VALIDATE(NUMERIC):Z2 :: GOT
O 500
700 IF PP=3 THEN GOSUB 660 :
: ACCEPT AT(5,15)SIZE(9)BEEP
 VALIDATE(NUMERIC):Z3 :: GOT
O 500
710 IF PP=4 THEN GOSUB 660 :
: ACCEPT AT(7,15)SIZE(9)BEEP
 VALIDATE(NUMERIC):Z4 :: GOT
O 500
720 IF PP=5 THEN GOSUB 660 :
: ACCEPT AT(9,15)SIZE(9)BEEP
 VALIDATE(NUMERIC):Z5 :: GOT
O 500
730 IF PP=6 THEN GOSUB 660 :
: ACCEPT AT(11,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z6 :: GO
TO 500
740 IF PP=7 THEN GOSUB 660 :
: ACCEPT AT(13,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z7 :: GO
TO 500
750 IF PP=8 THEN GOSUB 660 :
: ACCEPT AT(15,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z8 :: GO
TO 500
760 IF PP=9 THEN GOSUB 660 :
: ACCEPT AT(17,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z9 :: GO
TO 500
770 IF PP=10 THEN GOSUB 660
:: ACCEPT AT(19,15)SIZE(9)BE
EP VALIDATE(NUMERIC):Z10 ::
GOTO 500
780 DISPLAY AT(24,3):"f~p~m~
g~t~c~e~~help!h~" :: CALL KE
Y(0,K,S):: IF S=0 THEN 780 E
LSE 790
790 IF K=80 THEN 810 :: IF K
=72 THEN 1540 :: IF K=84 THE
N 410 :: IF K=71 THEN 970 ::
 IF K=77 THEN 1620 :: IF K=6
7 THEN 1580
800 IF K=69 THEN 370 :: IF K
=70 THEN 1670 ELSE 510
810 IF ST=0 THEN 820 ELSE 83
0
820 DISPLAY AT(24,3):"keine
werte angelegt" :: CALL KEY(
0,K,S):: IF S=0 THEN 820 ELS
E 510
830 X=100. :: DISPLAY AT(22,
25)SIZE(4):X :: DISPLAY AT(2
3,27)SIZE(1):"%" :: P1=Z1/ST
*X :: DISPLAY AT(1,25)SIZE(4
):P1 :: P2=Z2/ST*X
840 DISPLAY AT(3,25)SIZE(4):
P2 :: P3=Z3/ST*X :: DISPLAY
AT(5,25)SIZE(4):P3 :: P4=Z4/
ST*X :: DISPLAY AT(7,25)SIZE
(4):P4
850 P5=Z5/ST*X :: DISPLAY AT
(9,25)SIZE(4):P5 :: P6=Z6/ST
*X :: DISPLAY AT(11,25)SIZE(
4):P6 :: P7=Z7/ST*X :: DISPL
AY AT(13,25)SIZE(4):P7 :: P8
=Z8/ST*X
860 DISPLAY AT(15,25)SIZE(4)
:P8 :: P9=Z9/ST*X :: DISPLAY
 AT(17,25)SIZE(4):P9 :: P10=
Z10/ST*X :: DISPLAY AT(19,25
)SIZE(4):P10
870 CALL KEY(0,K,S):: IF S=0
 THEN 870 ELSE 510
880 CALL CLEAR :: OPEN #1:"C
S1",INTERNAL,SEQUENTIAL,OUTP
UT,FIXED 192 :: PRINT #1:TT,
MM,JJ :: PRINT #1:DN$
890 PRINT #1:F1$,F2$,F3$,F4$
,F5$,F6$,F7$,F8$,F9$,F10$
900 PRINT #1:Z1,Z2,Z3,Z4,Z5,
Z6,Z7,Z8,Z9,Z10
910 CLOSE #1 :: GOTO 370
920 CALL CLEAR :: ON ERROR 1
600
930 OPEN #1:"CS1",INTERNAL,S
EQUENTIAL,INPUT ,FIXED 192 :
: INPUT #1:TT,MM,JJ :: INPUT
 #1:DN$
940 INPUT #1:F1$,F2$,F3$,F4$
,F5$,F6$,F7$,F8$,F9$,F10$
950 INPUT #1:Z1,Z2,Z3,Z4,Z5,
Z6,Z7,Z8,Z9,Z10
960 CLOSE #1 :: GOTO 370
970 DISPLAY AT(1,1)SIZE(15):
"  5%" :: DISPLAY AT(3,1)SIZ
E(15):" 10%" :: DISPLAY AT(5
,1)SIZE(15):" 20%" :: DISPLA
Y AT(7,1)SIZE(15):" 30%"
980 DISPLAY AT(9,1)SIZE(15):
" 40%"
990 DISPLAY AT(11,1)SIZE(15)
:" 50%" :: DISPLAY AT(13,1)S
IZE(15):" 60%" :: DISPLAY AT
(15,1)SIZE(15):" 70%" :: DIS
PLAY AT(17,1)SIZE(15):" 80%"
1000 DISPLAY AT(19,1)SIZE(15
):" 90%"
1010 DISPLAY AT(21,1)SIZE(15
):"100%" :: DISPLAY AT(22,1)
SIZE(15):"....1234567890" ::
 DISPLAY AT(23,1)SIZE(15):".
.....1"
1020 FOR L=14 TO 14 :: CALL
COLOR(L,5,2):: CALL CHAR(136
,"FFFFFFFFFFFFFFFFFFFF"):: 
NEXT L
1030 FOR OI=13 TO 13 :: CALL
 COLOR(OI,14,2):: CALL CHAR(
128,"FFFFFFFFFFFFFFFFFFFF"):: 
```

```
 NEXT OI :: IF P1<1 THEN 108
0
1040 IF P1>1 AND P1<6 THEN A
=1 ELSE IF P1>5 AND P1<11 TH
EN A=3 ELSE IF P1>10 AND P3<
21 THEN A=5 ELSE IF P1>20 AN
D P1<31 THEN A=7
1050 IF P1>30 AND P1<41 THEN
 A=9 ELSE IF P1>40 AND P1<51
 THEN A=11 ELSE IF P1>50 AND
 P1<61 THEN A=13
1060 IF P1>60 AND P1<71 THEN
 A=15 ELSE IF P1>70 AND P1<8
1 THEN A=17 ELSE IF P1>80 AN
D P1<91 THEN A=19 ELSE IF P1
>91 THEN A=21
1070 CALL VCHAR(1,7,136,A)
1080 IF P2<1 THEN 1130
1090 IF P2>1 AND P2<6 THEN A
B=1 ELSE IF P2>5 AND P2<11 T
HEN AB=3 ELSE IF P2>10 AND P
2<21 THEN AB=5 ELSE IF P2>20
 AND P2<31 THEN AB=7
1100 IF P2>30 AND P2<41 THEN
 AB=9 ELSE IF P2>40 AND P2<5
1 THEN AB=11 ELSE IF P2>50 A
ND P2<61 THEN AB=13 ELSE IF
P2>60 AND P2<71 THEN AB=15
1110 IF P2>70 AND P2<81 THEN
 AB=17 ELSE IF P2>80 AND P2<
91 THEN AB=19 ELSE IF P2>91
THEN AB=21
1120 CALL VCHAR(1,8,128,AB)
1130 IF P3<1 THEN 1180
1140 IF P3>1 AND P3<6 THEN A
C=1 ELSE IF P3>5 AND P3<11 T
HEN AC=3 ELSE IF P3>10 AND P
3<21 THEN AC=5 ELSE IF P3>20
 AND P3<31 THEN AC=7
1150 IF P3>30 AND P3<41 THEN
 AC=9 ELSE IF P3>40 AND P3<5
1 THEN AC=11 ELSE IF P3>50 A
ND P7<61 THEN AC=13 ELSE IF
P3>60 AND P3<71 THEN AC=15
1160 IF P3>70 AND P3<81 THEN
 AC=17 ELSE IF P3>80 AND P3<
91 THEN AC=19 ELSE IF P3>91
THEN AC=21
1170 CALL VCHAR(1,9,136,AC)
1180 IF P4<1 THEN 1230
1190 IF P4>1 AND P4<6 THEN A
D=1 ELSE IF P4>5 AND P4<11 T
HEN AD=3 ELSE IF P4>10 AND P
4<21 THEN AD=5 ELSE IF P4>20
 AND P4<31 THEN AD=7
1200 IF P4>30 AND P4<41 THEN
 AD=9 ELSE IF P4>40 AND P4<5
1 THEN AD=11 ELSE IF P4>50 A
ND P4<61 THEN AD=13 ELSE IF
P4>60 AND P4<71 THEN AD=15
1210 IF P4>70 AND P4<81 THEN
 AD=17 ELSE IF P4>80 AND P4<
91 THEN AD=19 ELSE IF P4>91
THEN AD=21
1220 CALL VCHAR(1,10,128,AD)
1230 IF P5<1 THEN 1280
1240 IF P5>1 AND P5<6 THEN A
E=1 ELSE IF P5>5 AND P5<11 T
HEN AE=3 ELSE IF P5>10 AND P
5<21 THEN AE=5 ELSE IF P5>20
 AND P5<31 THEN AE=7
1250 IF P5>30 AND P5<41 THEN
 AE=9 ELSE IF P5>40 AND P5<5
1 THEN AE=11 ELSE IF P5>50 A
ND P5<61 THEN AE=13 ELSE IF
P5>60 AND P5<71 THEN AE=15
1260 IF P5>70 AND P5<81 THEN
 AE=17 ELSE IF P5>80 AND P5<
91 THEN AE=19 ELSE IF P5>91
THEN AE=21
1270 CALL VCHAR(1,11,136,AE)
1280 IF P6<1 THEN 1330
1290 IF P6>1 AND P6<6 THEN A
F=1 ELSE IF P6>5 AND P6<11 T
HEN AF=3 ELSE IF P6>10 AND P
6<21 THEN AF=5 ELSE IF P6>20
 AND P6<31 THEN AF=7
1300 IF P6>30 AND P6<41 THEN
 AF=9 ELSE IF P6>40 AND P6<5
1 THEN AF=11 ELSE IF P6>50 A
ND P6<61 THEN AF=13 ELSE IF
P6>60 AND P6<71 THEN AF=15
1310 IF P6>70 AND P6<81 THEN
 AF=17 ELSE IF P6>80 AND P6<
91 THEN AF=19 ELSE IF P6>91
THEN AF=21
1320 CALL VCHAR(1,12,128,AF)
1330 IF P7<1 THEN 1380
1340 IF P7>1 AND P7<6 THEN A
G=1 ELSE IF P7>5 AND P7<11 T
HEN AG=3 ELSE IF P7>10 AND P
7<21 THEN AG=5 ELSE IF P7>20
 AND P7<31 THEN AG=7
1350 IF P7>30 AND P7<41 THEN
 AG=9 ELSE IF P7>40 AND P7<5
1 THEN AG=11 ELSE IF P7>50 A
ND P7<61 THEN AG=13 ELSE IF
P7>60 AND P7<71 THEN AG=15
1360 IF P7>70 AND P7<81 THEN
 AG=17 ELSE IF P7>80 AND P7<
91 THEN AG=19 ELSE IF P7>91
THEN AG=21
1370 CALL VCHAR(1,13,136,AG)
1380 IF P8<1 THEN 1430
1390 IF P8>1 AND P8<6 THEN A
H=1 ELSE IF P8>5 AND P8<11 T
HEN AH=3 ELSE IF P8>10 AND P
8<21 THEN AH=5 ELSE IF P8>20
 AND P8<31 THEN AH=7
1400 IF P8>30 AND P8<41 THEN
 AH=9 ELSE IF P8>40 AND P8<5
```

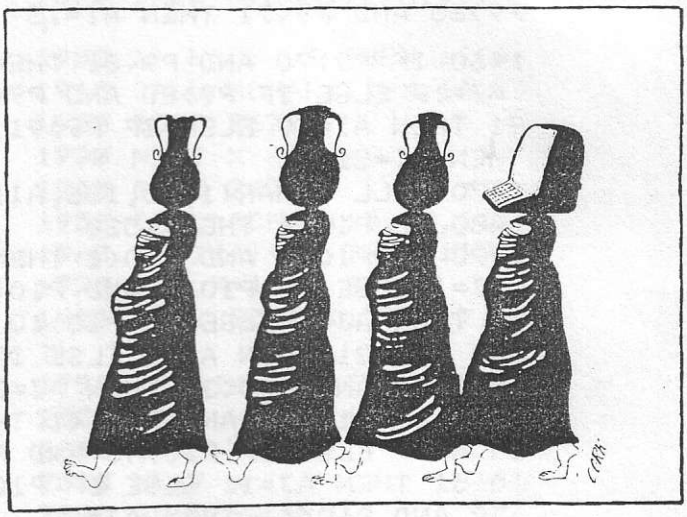
```
i THEN AH=11 ELSE IF P8>50 A
ND P8<61 THEN AH=13 ELSE IF
P8>60 AND P8<71 THEN AH=15
1410 IF P8>70 AND P8<81 THEN
 AH=17 ELSE IF P8>80 AND P8<
91 THEN AH=19 ELSE IF P8>91
THEN AH=21
1420 CALL VCHAR(1,14,128,AH)
1430 IF P9<1 THEN 1480
1440 IF P9>1 AND P9<6 THEN A
I=1 ELSE IF P9>5 AND P9<11 T
HEN AI=3 ELSE IF P9>10 AND P
9<21 THEN AI=5 ELSE IF P9>20
 AND P9<31 THEN AI=7
1450 IF P9>30 AND P9<41 THEN
 AI=9 ELSE IF P9>40 AND P9<5
1 THEN AI=11 ELSE IF P9>50 A
ND P9<61 THEN AI=13 ELSE IF
P9>60 AND P9<71 THEN AI=15
1460 IF P9>70 AND P9<81 THEN
 AI=17 ELSE IF P9>80 AND P9<
91 THEN AI=19 ELSE IF P9>91
THEN AI=21
1470 CALL VCHAR(1,15,136,AI)
1480 IF P10<1 THEN 1530
1490 IF P10>1 AND P10<6 THEN
 AJ=1 ELSE IF P10>5 AND P10<
11 THEN AJ=3 ELSE IF P10>10
AND P10<21 THEN AJ=5 ELSE IF
 P10>20 AND P10<31 THEN AJ=7
1500 IF P10>30 AND P10<41 TH
EN AJ=9 ELSE IF P10>40 AND P
10<51 THEN AJ=11 ELSE IF P10
>50 AND P10<61 THEN AJ=13
1510 IF P10>60 AND P10<71 TH
EN AJ=15 ELSE IF P10>70 AND
P10<81 THEN AJ=17 ELSE IF P1
0>80 AND P10<91 THEN AJ=19 E
LSE IF P10>91 THEN AJ=21
1520 CALL VCHAR(1,16,128,AJ)
1530 CALL KEY(0,K,S):: IF S=
0 THEN 1530 ELSE 780
1540 CALL CLEAR :: DISPLAY A
T(1,1);">G GRAPHIK" :: DISPL
AY AT(3,4):"PRO POS.IN %" ::
 DISPLAY AT(5,1):">T TABELLE
"
1550 DISPLAY AT(7,1):">P %AU
SWERTUNG" :: DISPLAY AT(9,1)
:">M MITTELWERT" :: DISPLAY
AT(11,1):">C HARDCOPY" :: DI
SPLAY AT(13,1):">E ENDE"
1560 DISPLAY AT(15,1):">F BE
R.NACH FORMELN"
1570 CALL KEY(0,K,S):: IF S=
0 THEN 1570 ELSE 370
1580 ON ERROR 1600 :: OPEN #
1:DR$,OUTPUT :: FOR Y=1 TO 2
4 :: Q$="" :: FOR X=32 TO 1
STEP -1 :: CALL GCHAR(Y,X,WE
):: WT$=CHR$(WE):: Q$=WT$+Q$
1590 NEXT X :: PRINT #1:TAB(
22);Q$ :: NEXT Y :: CLOSE #1
 :: GOTO 370
1600 DISPLAY AT(5,1)ERASE AL
L:"DATA ODER FILE ERROR !" :
: DISPLAY AT(9,1):"PROGRAMM
ABBRECHEN/NEUSTART!"
1610 CALL KEY(0,K,S):: IF S=
0 THEN 1610 ELSE 370
1620 IF Z1>0 THEN P01=Z1-Z1+
1 :: IF Z2>0 THEN P02=Z2-Z2+
1 :: IF Z3>0 THEN P03=Z3-Z3+
1 :: IF Z4>0 THEN P04=Z4-Z4+
1 :: IF Z5>0 THEN P05=Z5-Z5+
1
1630 IF Z6>0 THEN P06=Z6-Z6+
1 :: IF Z7>0 THEN P07=Z7-Z7+
1 :: IF Z8>0 THEN P08=Z8-Z8+
1 :: IF Z9>0 THEN P09=Z9-Z9+
1 :: IF Z10>0 THEN P010=Z10-
Z10+1
1640 PT=P01+P02+P03+P04+P05+
P06+P07+P08+P09+P010
1650 MW=ST/PT :: DISPLAY AT(
24,3):"~mittelwert~~" :: DIS
PLAY AT(24,16):USING "######
.##":MW
1660 CALL KEY(0,K,S):: IF S=
0 THEN 1660 :: IF K=13 THEN
780 ELSE 1660
1670 FOR OI=13 TO 13 :: CALL
 COLOR(OI,14,2):: CALL CHAR(
128,"FFFFFFFFFFFFFFFF"):: N
EXT OI
1680 CALL CLEAR :: CALL VCHA
R(1,28,128,12):: CALL VCHAR(
1,29,128,12):: CALL VCHAR(1,
30,128,12):: DISPLAY AT(1,26
):"tan"
1690 DISPLAY AT(3,26):"cos"
:: DISPLAY AT(5,26):"log" ::
 DISPLAY AT(7,26):"atn" :: D
ISPLAY AT(9,26):"sin" :: DIS
PLAY AT(11,26):"sqr"
1700 DISPLAY AT(24,1):"auswa
hl~¦~~t~c~l~a~s~q~~"
1710 CALL KEY(0,K,S):: IF S=
0 THEN 1710 :: IF K=84 THEN
1730 :: IF K=67 THEN 1770 ::
 IF K=76 THEN 1790
1720 IF K=65 THEN 1810 :: IF
 K=83 THEN 1830 :: IF K=81 T
HEN 1850 ELSE 1710
1730 DISPLAY AT(1,25)SIZE(1)
:"X" :: GOSUB 1870
1740 DISPLAY AT(9,1):TAN(AL)
:: DISPLAY AT(11,1):TAN(BE):
: DISPLAY AT(13,1):"GRAD";TA
N(GA*PI/180)
1750 DISPLAY AT(24,1):"h~cop
```

```
y!c^help!h^ende!e^"
1760 CALL KEY(0,K,S):: IF S=
0 THEN 1760 :: IF K=67 THEN
1580 :: IF K=72 THEN 1540 ::
 IF K=69 THEN 410 ELSE 1670
1770 DISPLAY AT(3,25)SIZE(1)
:"X" :: GOSUB 1870
1780 DISPLAY AT(9,1):COS(AL)
:: DISPLAY AT(11,1):COS(BE):
: DISPLAY AT(13,1):"GRAD";CO
S(GA*PI/180):: GOTO 1750
1790 DISPLAY AT(5,25)SIZE(1)
:"X" :: DISPLAY AT(3,1)SIZE(
4):"WERT" :: ON WARNING NEXT
 :: ACCEPT AT(3,13)SIZE(10)V
ALIDATE(DIGIT,"."):WRT
1800 DISPLAY AT(5,1)SIZE(12)
:"LOGARITHMUS=" :: DISPLAY A
T(7,1):LOG(WRT):: GOTO 1750
1810 DISPLAY AT(7,25)SIZE(1)
:"X" :: GOSUB 1870
1820 DISPLAY AT(11,1):ATN(AL
):: DISPLAY AT(13,1):ATN(BE)
:: DISPLAY AT(15,1):"GRAD";A
TN(GA*180/PI):: GOTO 1750
1830 DISPLAY AT(9,25)SIZE(1)
:"X" :: GOSUB 1870
1840 DISPLAY AT(11,1):SIN(AL
):: DISPLAY AT(13,1):SIN(BE)
:: DISPLAY AT(15,1):"GRAD";S
IN(GA*PI/180):: GOTO 1750
1850 DISPLAY AT(11,25)SIZE(1
):"X" :: DISPLAY AT(3,1)SIZE
(4):"WERT" :: ON WARNING NEX
T :: ACCEPT AT(3,13)SIZE(10)
VALIDATE(DIGIT,"."):WERT
1860 DISPLAY AT(5,1)SIZE(14)
:"QUADRATWURZEL=" :: DISPLAY
 AT(7,1):SQR(WERT):: GOTO 17
50
1870 DISPLAY AT(3,1)SIZE(11)
:"BOGENMASS 1" :: ON WARNING
 NEXT :: ACCEPT AT(3,13)SIZE
(10)VALIDATE(DIGIT,"."):AL
1880 DISPLAY AT(5,1)SIZE(11)
:"BOGENMASS 2" :: ACCEPT AT(
5,13)SIZE(10)VALIDATE(DIGIT,
"."):BE :: DISPLAY AT(7,1)SI
ZE(8):"GRADMASS"
1890 ACCEPT AT(7,13)SIZE(10)
VALIDATE(DIGIT,"."):GA
1900 RETURN
```

Die nächste TI REVUE erscheint am 28. Oktober

SERVICE

FÜR BUCHHALTER MUSIKER & SAMMLER



SPICKZETTEL

Nachdem das Programm mit RUN, RUN" DSK1. filename" oder RUN" CS1" gestartet worden ist, erscheint das Titelbild. Dazu ertönt die etwas verfremdete Melodie des Liedes „Yellow Submarine" von den Beatles. Mit einer beliebigen Taste geht es weiter. Man wird nun nach dem Namen gefragt und ob der Drukker angeschlossen ist, um das Zeugnis auszudrukken. Jetzt kann man noch wählen, ob man die Spielanleitung lesen will. Nach all diesen Vorbereitungen beginnt nun endlich das eigentlich Spiel.

Obwohl der arme Schüler Willi weiß, daß heute einige Exen (= Stegreifaufgaben) ins Haus stehen, hat er wie üblich nichts gelernt. Als sein Schutzengel steigst Du nun in Willis Körper. Dadurch ist er zwar auch nicht schlauer, aber Du kannst ihm helfen, Spickpunkte einzusammeln, die von Zeit zu Zeit auf einer Schulbank erscheinen. Das versucht der Lehrer natürlich zu verhindern, indem er ihn verfolgt und Sehstrahlen nach ihm wirft. Wird Willi von einem Sehstrahl getroffen, hat ihn der Lehrer beim Spicken erwischt und Du erntest eine Spicksechs. Du bist aber auch nicht ganz wehrlos, Du kannst nämlich mit U-Haken auf den Lehrer schießen. Triffst Du ihn, ist er fünf Minu-

ten bzw. bis zum nächsten gefressenen Spickpunkt bewußtlos. Danach ist er allerdings noch zorniger, d.h. er fährt schneller und schießt öfter. Hast Du 45 (Spiel-) Minuten überstanden oder bist Du vom Lehrer erwischt worden, ist die erste Stunde zu Ende, und der Computer errechnet die Zeugnisnote aus den erreichten Spickpunkten (wenn man nicht ohnehin schon eine Sechs hat). Jetzt folgt die zweite Schulstunde mit einem anderen Fach. Die Lehrer werden von Stunde zu Stunde strenger, was sich in ihrer Geschwindigkeit und der Häufigkeit der Sehstrahlen zeigt. Sind alle sieben Stunden vorbei, wird aus der Summe der abgegebenen Schüsse und der Zahl der Spicksechser die Betragennote errechnet. Anschließend wird das Zeugnis ausgegeben (wenn der Drucker angeschlossen ist, auch schwarz auf weiß). Durch Tastendruck erscheint die Frage, ob man noch einmal spielen will. Betätigt man die Taste „J" beginnt das Spiel von vorne, aber ohne Titelbild.

Noch ein paar Hinweise zum Gebrauch des Programms:

1. Gesteuert wird Willi mit dem Joystick 1. Die Uhaken feuert man mit dem Feuerknopf des Joysticks ab.
2. Während des Lehrerwechsels können über die Funktionstasten noch weitere Funktionen betätigt werden:
   mit „CLEAR" kommt man zur Frage, ob man aufhören möchte.
   mit „AID" kommt man zum Zeugnis.
   mit „BACK" beginnt das Spiel von vorne.

Peter Hlauna

Musikkompositionsprogramm

Dieses Programm ist ganz der Musik gewidmet! Es bietet dem Benutzer die Möglichkeit der Eingabe einer Melodie mit bis zu 125 Noten, wobei ihm die Auswahl zwischen Ein-, Zwei- oder Dreiklang angeboten wird. Die eingegebene Melodie kann nach einer eventuellen Korrektur beliebig oft angehört werden. Zusätzlich besteht die Möglichkeit, die Komposition auf Cassette zu speichern oder andere zu laden. Die im Menü angebotene Einleitung (Wahlnummer 1) gibt Einzelinformationen zur Programmhandhabung.

Bei Titelbild und Schlußbild setzt das Drücken einer beliebigen Taste das Programm fort.

Benötigt werden für das Programm nur die Konsole, ein Datenrecorder und Spaß am TI.

P.S. Dieses Programm darf nur mit der Konsole, ohne Xbasic-Modul und ohne angeschlossene Peripherie gefahren werden, da zur Dimensionierung der Felder fast der gesamte Speicherplatz benötigt wird.

Glückliche Besitzer einer 32-K-Erweiterung können das Programm jedoch auch in Extended-Basic fahren.

Alexander Saberschinsky

Steuerberechnung

Das Programm unterscheidet zwischen sozialversicherungs- und nicht sozialversicherungspflichtigen Arbeitnehmern und errechnet nach Eingabe von Bruttojahreseinkommen, Werbungskosten, Sonderausgaben, Zahl der Kinder und etwaiger Baukosten für ein Eigenheim oder eine Eigentumswohnung das zu versteuernde Einkommen, die jährliche Einkommensteuer, die jährliche Kirchensteuer sowie monatliche Arbeitnehmerbeiträge zur Kranken-, Renten- und Arbeitslosenversicherung und das monatliche Nettoeinkommen unter Berücksichtigung einer etwaigen Steuerersparnis nach § 7b EStG. Sollten die genauen Werte für Werbungskosten und Sonderausgaben noch nicht bekannt sein, zeigt das Programm die familienstandabhängigen Pauschalbeträge für diese Werte extra an. Nach Abschluß jeder Berechnung kann zu einer Druckroutine verzweigt werden, die die Werte „schwarz auf weiß" festhält.

Das Programm ist nicht nur für Steuerberater bzw. Vermögensberater interessant, sondern auch für jeden Arbeitnehmer, der über seine Abgaben Bescheid wissen möchte bzw. errechnen möchte, ob etwa eine Erhöhung der Sonderausgaben über z.B. Lebensversicherung o.ä. zu einer Steuerersparnis führt etc.

Bei der Eingabe der Sozialversicherungsquoten am Anfang des Programms ist es wichtig, nur die Arbeitnehmeranteile in % einzugeben.

Thomas Ehlert

Advantage

Bei dem Spiel Advantage werden verschiedene farbige Kugeln in ein Labyrinth von beweglichen Hebeln geworfen, die je nach ihrer Stellung die Kugeln festhalten bzw. freigeben.

Jeder Spieler muß eine vorgegebene Anzahl von Kugeln je Farbe durch das Labyrinth bringen.

Der Spieler, dessen Kugeln alle das Ende des Labyrinth zuerst erreicht haben, ist Sieger.

TI-Fall:

Herzlich willkommen zu TI-Fall, einer mehr oder weniger gelungenen Nachahmung des berühmten Pitfall. Zu Beginn meiner Version von Pitfall werden Sie mit den Rhythmen von La Cucaracha gleich in die Urwaldatmosphäre eingestimmt.

Und nachdem es sich um eine Nachahmung handelt, ist auch hier das Ziel des Spielers, möglichst viele Bilder unbeschadet zu durchlaufen und dabei so viel Punkte zu sammeln, wie nur möglich. Punkte liegen dabei in Form von Goldbarren umher und um diese aufzusammeln, braucht man nur mit seinem Männchen, genannt Cuthbert, darüberzulaufen.

Daß das Ganze natürlich nicht so einfach ist, versteht sich eigentlich von alleine. Lagerfeuer, Baumstämme, Skorpione, Tümpel und gefährliche Vögel bilden den Reiz des Spieles.

Mit ihrem Joystick haben Sie nun die Möglichkeit, ihren Cuthbert nach links oder rechts laufen zu lassen oder ihn mit dem Fire Button Hindernisse überspringen zu lassen.

Leider ist das Listing nicht besonders übersichtlich und daher nicht so einfach einzutippen, jedoch sollte Sie der Spielspaß dafür entschädigen.

Michael Schneller

```
10 ! **********************
11 ! *                    *
12 ! *      TI-FALL       *
13 ! *                    *
14 ! *    Copyright by    *
15 ! * Michael Schneller  *
16 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *    Joystick (1)    *
23 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     9093 Bytes     *
28 ! *                    *
29 ! **********************
30 !
100 CALL CHAR(34,"FEFDFBF7EF
DFBF7F7FBFDFEFF7FBFDFE")
110 A$=RPT$("F",16):: CALL C
HAR(112,A$,96,A$,97,"0303030
303030303"):: CALL COLOR(9,1
2,2,10,12,2,11,13,11,1,11,1)
120 CALL CHAR(113,""):: RAND
OMIZE :: CALL SCREEN(2):: CA
LL CLEAR :: CALL MAGNIFY(3)
130 CALL CHAR(100,"0C0C001C1
C3838180018141 4E2820203"&RPT
$("0",32),104,"0C0C000C1C1C1
C1C000C0E0A3E28080C"&RPT$("0
",32))
140 CALL CHAR(108,"181800183
C3A3A1800186483000000"&RPT$(
"0",32))
150 CALL CHAR(116,RPT$("0",1
8)&"708D0303038D72"&RPT$("0"
,17)&"C02F1F9FEF0E01")
160 CALL CHAR(120,RPT$("0",1
6)&"30408F9F7F0F0708"&RPT$("
0",19)&"EB1C0C0C0B14E")
170 CALL CHAR(124,RPT$("F",6
4),128,"C0F0FC"&RPT$("F",20)
&"FCF0C000000000C0F0FCFEFEFC
F0C")
180 CALL CHAR(132,"000000000
30F3F7F7F3F0F0300000000030F3
F"&RPT$("F",20)&"3F0F03")
190 CALL CHAR(136,RPT$("0",1
6)&"01071FFEFE1F0701"&RPT$("
0",16)&"80E0F87F7FF8E08")
200 CALL CHAR(48,"00FEC6C6C6
C6C6FE",49,"0018181818181818
",50,"00FE0606FEC0C0FE",51,"
00FE06063E0606FE")
210 CALL CHAR(52,"00D8D8D8FE
181818",53,"00FEC0C0FE0606FE
",54,"00C0C0C0FEC6C6FE",55,"
00FE06063E060606")
220 CALL CHAR(56,"00FEC6C6FE
C6C6FE",57,"00FEC6C6FE060606
")
230 RESTORE 1020 :: FOR I=65
 TO 90 :: READ C$ :: CALL CH
AR(I,C$):: NEXT I
240 CALL CHAR(92,"0000000000
30CD0303010000000000000000000
00000CB3C0C08")
250 RESTORE 1160 :: FOR I=1
TO 5 :: READ V(I):: NEXT I
260 CALL CHAR(33,"FF818181FF
818181",140,"0103070F0F0F0F0
703000102040402 01C0E0B0D8D8B
8F8F0E08")
270 DIM SPS(91),LL(91):: HI=
0 :: M(1)=97 :: M(2)=225 ::
DEF SKS=INT(200*RND)+40
280 CALL CHAR(42,"1082105400
FEFEFEFF818181FF818181"):: C
ALL COLOR(2,16,11)
290 RESTORE 1080 :: FOR LI=1
 TO 91 :: READ SPS(LI),L ::
LL(LI)=1000/L :: NEXT LI
300 CALL CHAR(44,"0000010103
0303070D1E3F767E7D3F1F008080
C0C0C0A0F0F8AC7CBECEFEBCF8")
310 CALL VORSPANN(SPS(),LL()
):: MA=4 :: \=81 :: _=-1 ::
CALL CLEAR :: CALL BAUM :: C
ALL HCHAR(24,15,100,MA-1)::
[(1)=100 :: [(3)=104
320 DISPLAY AT(24,19):"HI:";
HI :: GOTO 600
330 Q=1 :: DISPLAY AT(24,19)
:"HI:";HI
340 CALL JOYST(1,X,Y):: _=-_
 :: CALL KEY(1,K,S):: IF Y=4
 THEN 520 ELSE CALL MOTION(#
1,0,X*1.5):: IF K=18 THEN 81
0 ELSE CALL POSITION(#1,Y1,N
)
350 IF N<15 THEN 510 ELSE IF
 N>240 THEN 600 ELSE CALL PA
TTERN(#1,[(_+2)):: CALL GCHA
R(((\+7)/8)+1,((N+7)/8),H)::
 IF H=32 THEN 540 ELSE IF H=
42 THEN 570
360 ON Q GOTO 340,380,420,43
0,490,450,420
370 GOTO 340
380 C=C+1 :: CALL COINC(ALL,
E):: IF E THEN 910 ELSE IF C
=4 THEN 390 ELSE 370
390 IF O=0 THEN CALL SPRITE(
#2,124,5,81,113,#3,128,5,81,
129,#4,132,5,81,97):: C=0 EL
SE CALL DELSPRITE(#3,#2,#4):
: C=0
400 IF O=0 THEN O=1 ELSE O=0
410 GOTO 370
420 CALL COINC(ALL,E):: IF E
 THEN 910 ELSE 340
```

```
430 CALL POSITION(#2,Y2,X2):
: CALL MOTION(#2,0,SGN(N-X2)
*5):: IF SGN(N-X2)=-1 THEN C
ALL PATTERN(#2,116)ELSE CALL
 PATTERN(#2,120)
440 GOTO 420
450 FE=-FE :: ON FE+2 GOTO 4
60,460,470
460 CALL LOCATE(#2,73,80)::
GOTO 480
470 CALL LOCATE(#2,193,1)
480 GOTO 420
490 ZV=ZV+1 :: IF ZV>5 THEN
ZV=0
500 CALL MOTION(#11,V(ZV),-1
0):: GOTO 420
510 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,\,20):: GOTO 34
0
520 CALL MOTION(#1,0,0):: CA
LL GCHAR(INT((\+7)/8)-2,INT(
(N+7)/8),H):: IF H=33 THEN 5
30 ELSE 360
530 FOR @=153 TO 81 STEP -9
:: CALL LOCATE(#1,@,121):: C
ALL SOUND(-100,@+100,0):: NE
XT @ :: \=81 :: GOTO 360
540 CALL MOTION(#1,0,0):: CA
LL MOTION(#1,30,0):: FOR @=1
 TO 100 :: NEXT @ :: CALL MO
TION(#1,0,0):: CALL LOCATE(#
1,153,N)
550 \=153 :: CALL SOUND(-100
,-8,0,220,30,220,30,330,30):
: GOTO 340
560 GOTO 360
570 CALL MOTION(#1,0,0,#2,0,
0,#11,0,0):: FOR SOU=0 TO 30
 STEP 2 :: CALL SOUND(-100,3
80,SOU,640,SOU):: NEXT SOU
580 FOR @=1 TO 10 :: P=P+100
 :: CALL PU(P):: CALL SOUND(
-50,843,1):: NEXT @
590 CALL HCHAR(((\+7)/8)+1,(
(N+7)/8),113):: CALL PU(P)::
 GOTO 340
600 CALL DELSPRITE(ALL):: FO
R S=0 TO 30 STEP 2 :: CALL S
OUND(-100,262,S,330,S,392,S)
:: NEXT S
610 P=P+100 :: CALL PU(P)::
Q=INT(7*RND)+1
620 CALL HCHAR(10,1,113,128)
:: CALL VCHAR(15,16,32,4)::
CALL HCHAR(19,1,113,128)
630 ON Q GOSUB 650,660,670,7
10,800,750,780
640 CALL SPRITE(#1,100,5,\,2
0):: GOTO 340
650 CALL LOCH(1):: CALL BAR(
2):: RETURN
660 O,C=0 :: CALL BAR(1):: R
ETURN
670 CALL LOCH(1):: G=INT(2*R
ND)+1 :: FOR @=1 TO G :: CAL
L SPRITE(#@+1,136,12,81,M(@)
):: NEXT @
680 D=INT(2*RND)+1 :: FOR @=
1 TO D :: CALL SPRITE(#@+4,1
36,12,153,M(@)):: NEXT @
690 R=INT(7*RND)-12 :: R1=IN
T(5*RND)-13 :: FOR @=1 TO G
:: CALL MOTION(#@+1,0,R):: N
EXT @
700 FOR @=1 TO D :: CALL MOT
ION(#@+4,0,R1):: NEXT @ :: R
ETURN
710 CALL LOCH(1):: @=INT(2*R
ND):: IF @=1 THEN CALL SPRIT
E(#2,116,16,81,SKS,0,-5)ELSE
 CALL SPRITE(#2,116,16,153,S
KS,0,-5)
720 IF @=1 THEN YG=153 ELSE
YG=81
730 U=INT(2*RND)+1 :: FOR @=
1 TO U :: CALL SPRITE(#@+2,1
36,12,YG,M(@)):: NEXT @ :: J
=INT(5*RND)-13
740 FOR @=1 TO U :: CALL MOT
ION(#@+2,0,J):: NEXT @ :: RE
TURN
750 CALL BAR(2):: CALL HCHAR
(12,11,34):: CALL HCHAR(12,1
2,35):: CALL SPRITE(#2,44,7,
73,80):: CALL LOCH(2)
760 @=INT(2*RND):: IF @=0 TH
EN FE=-1 ELSE FE=1
770 IF RND>.5 THEN CALL SPRI
TE(#3,136,2,153,240,0,INT(6*
RND)-12):: RETURN ELSE RETUR
N
780 CALL SPRITE(#2,124,5,81,
113,#3,132,5,81,97,#4,128,5,
81,129,#10,140,7,65,97,0,9)
790 CALL BAR(1):: RETURN
800 ZV=0 :: CALL SPRITE(#11,
92,16,65,233):: CALL BAR(1):
: RETURN
810 CALL MOTION(#11,0,-7)::
CALL SOUND(50,380,0):: CALL
PATTERN(#1,108)
820 FOR @=-18 TO 18 STEP 6 :
: CALL MOTION(#1,@,X*2.5)::
CALL POSITION(#1,Y1,N):: IF
N<10 THEN 510 ELSE IF N>240
THEN 600
830 CALL COINC(ALL,E):: IF E
 THEN 850
840 NEXT @ :: CALL MOTION(#1
,0,X*2.5):: CALL LOCATE(#1,\
,N):: GOTO 350
```

```
850 CALL MOTION(#1,0,0,#10,0
,0):: CALL COINC(#1,#10,15,E
):: IF E=0 THEN 910
860 CALL MOTION(#1,0,9,#10,0
,9)
870 FOR S=1 TO 90 :: CALL SO
UND(LL(S),SPS(S),0):: CALL K
EY(1,K,Z):: IF K=18 THEN 890
880 NEXT S :: GOTO 870
890 CALL SOUND(-100,392,0)::
 CALL MOTION(#1,0,0):: CALL
POSITION(#1,Y1,N):: CALL LOC
ATE(#1,\,N)
900 GOTO 360
910 CALL MOTION(#1,0,0):: CA
LL DELSPRITE(#2,#3,#4,#5,#6,
#7,#8):: FOR I=1 TO 10 :: CA
LL SOUND(-10,262,0)
920 CALL SOUND(-10,523,0)::
NEXT I :: CALL MOTION(#1,120
,0)
930 CALL POSITION(#1,Y1,N)::
 IF Y1<185 THEN 930
940 CALL DELSPRITE(#1):: FOR
 @=0 TO 30 STEP 2 :: CALL SO
UND(-50,-6,@):: NEXT @
950 MA=MA-1 :: CALL HCHAR(24
,15,32,3):: IF MA=0 THEN 970
 ELSE CALL HCHAR(24,15,100,M
A-1)
960 GOTO 600
970 CALL DELSPRITE(ALL):: DI
SPLAY AT(17,7):"G A M E   O
V E R" :: IF P>HI THEN HI=P
:: DISPLAY AT(24,19):"HI:";H
I
980 CALL VCHAR(15,16,32,4)
990 FOR S=1 TO 91 :: CALL SO
UND(LL(S),SPS(S),0):: CALL K
EY(0,K,Z):: IF Z=0 THEN
1000 ELSE 1010
1000 NEXT S :: GOTO 990
1010 CALL CLEAR :: P=0 :: GO
TO 310
1020 DATA 00FEC6C6C6FEC6C6,0
0FEC6C6F8C6C6FE,00FEC0C0C0C0
C0FE,00F8C6C6C6C6C6F8,00FEC0
C0F8C0C0FE
1030 DATA 00FEC0C0F0C0C0C0,0
0FEC0C0DEC6C6FE,00C6C6C6FEC6
C6C6,0018181818181818,00FE06
0606C6C6FE
1040 DATA 00CCCCCCCF0CCCCCCC,0
0C0C0C0C0C0C0FE,00FED6D6C6C6
C6C6,00F6F6F6F6CECECE,00FEC6
C6C6C6C6FE
1050 DATA 00FEC6C6FEC0C0C0,0
0CCCCCCCCCCEFD,00FEC6C6FEE0D8
C6,00FEC0C0FE0606FE,00FE1818
18181818
1060 DATA 00C6C6C6C6C6C6FE,0
0666666666181818,00C6C6C6D6D6
D6FE,00C6C6C638C6C6C6
1070 DATA 00C6C6FE30303030,0
0FE0606380C0FE

1080 DATA 131,8,131,8,131,8,
175,8,30000,4,220,8,30000,8,
131,8,131,8,131,8
1090 DATA 175,8,30000,4,220,
8,30000,2,30000,8,175,4,175,
8,165,8,165,8,147,8,147,8

1100 DATA 131,2,30000,8,131,
8,131,8,131,8,165,8,30000,4,
196,8,30000,8,131,8,131,8,13
1,8
1110 DATA 165,8,30000,4,196,
8,30000,2,30000,8,262,4,294,
8,262,8,233,8,220,8,196,8
1120 DATA 220,3,175,8,175,8,
131,4,131,8,175,8,175,8,220,
8,220,8,262,4,220,8,220,2
1130 DATA 30000,8,262,4,294,
8,262,8,233,8,220,8,262,8,23
3,3,196,8,196,2,30000,8,131,
4
1140 DATA 131,8,165,8,165,8,
196,8,196,8,233,3,196,8,196,
2,30000,8,233,4,233,8,262,8,
233,8,220,8
1150 DATA 196,8,220,3,175,8,
175,2,30000,8,220,3,175,8,17
5,8,175,8
1160 DATA 10,5,0,-5,-10
1170 SUB BAUM
1180 FOR @=3 TO 21 STEP 9 ::
 DISPLAY AT(1,@):"  ppp" ::
DISPLAY AT(2,@):" pppppp"
1190 DISPLAY AT(3,@):"pppppp
pp" :: DISPLAY AT(4,@):" ppp
ppp" :: DISPLAY AT(5,@):"  p
ppp"
1200 DISPLAY AT(6,@):"...''a
" :: DISPLAY AT(7,@):"...'''
" :: DISPLAY AT(8,@):"..''''"
 :: DISPLAY AT(9,@):"''''''''
"
1210 NEXT @ :: CALL HCHAR(10
,1,113,128):: CALL HCHAR(14,
1,112,32):: CALL HCHAR(19,1,
113,128)
1220 CALL HCHAR(23,1,112,32)
:: SUBEND
1230 SUB TI(Z):: FOR @=1 TO
Z :: NEXT @ :: SUBEND
1240 SUB PU(P):: DISPLAY AT(
24,1)SIZE(7):P :: SUBEND
1250 SUB LOCH([])
1260 ON [] GOTO 1270,1280,12
90
1270 @=INT(2*RND):: IF @=1 T
HEN CALL HCHAR(12,8,32,2)::
```

```
CALL HCHAR(13,8,32,2)
1280 @=INT(2*RND):: IF @=1 T
HEN CALL HCHAR(12,15,32,3)::
 CALL HCHAR(13,15,32,3):: CA
LL VCHAR(15,16,33,4):: CALL
HCHAR(13,16,33)
1290 @=INT(2*RND):: IF @=1 T
HEN CALL HCHAR(12,25,32,2)::
 CALL HCHAR(13,25,32,2)
1300 SUBEND
1310 SUB VORSPANN(SPS(),LL()
)
1320 CALL BAUM :: CALL HCHAR
(19,1,32,160):: CALL VCHAR(1
7,4,96,5)
1330 CALL HCHAR(17,2,96,5)::
 CALL VCHAR(17,8,96,5):: CAL
L HCHAR(19,10,96,2):: CALL V
CHAR(17,13,96,5)
1340 CALL HCHAR(17,14,96,3):
: CALL HCHAR(19,14,96,2):: C
ALL VCHAR(17,18,96,5):: CALL
 HCHAR(17,19,96,2):: CALL VC
HAR(17,21,96,5)
1350 CALL HCHAR(20,19,96,2):
: CALL VCHAR(17,23,96,5):: C
ALL HCHAR(21,24,96,3):: CALL
 VCHAR(17,28,96,5)
1360 CALL HCHAR(21,29,96,3):
: DISPLAY AT(24,1):"(C)1985
BY MICHAEL SCHNELLER"
1370 FOR @=3 TO 8 :: CALL CO
LOR(@,16,1):: NEXT @
1380 FOR S=1 TO 90 :: CALL S
OUND(LL(S),SPS(S),0):: CALL
KEY(0,K,Z):: IF Z=0 THEN 139
0 ELSE SUBEXIT
1390 NEXT S :: GOTO 1380
1400 SUBEND
1410 SUB BAR(\):: ON \ GOTO
1420,1430
1420 @=INT(5*RND):: IF @>2 T
HEN CALL HCHAR(21,20,42)
1430 @=INT(5*RND):: IF @>2 T
HEN CALL HCHAR(12,23,42)
1440 SUBEND
```


UNSER TELEFONSERVICE

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite

ACHTUNG !!!

Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13


```
10 ! **********************
11 ! *                    *
12 ! *    SPICKZETTEL     *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *    Peter Hlawna    *
16 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *    Joystick (1)    *
23 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *    11576 Bytes     *
28 ! *                    *
29 ! **********************
30 !
400 RANDOMIZE :: Q=1 :: GES=
3 :: SCHWER=.08 :: ZEIT,COUN
T,P,RU,SCH=0 :: FACH$="RELIG
ION"
410 !
420 ! ZEICHENDATAS
430 !
440 DATA 030F1F312E0E6EF1,BE
DD5D0E07020206,C0F0F88C74707
68F,7DB9BB72E24060
450 DATA 030F1F312E0E6EF1,BE
9DDD4E47020600,C0F0F88C74707
68F,7DBBBA70E0404060
460 DATA 0F1F30203F292620,21
200113100C03,F8FC0602FE4A328
2,4202C0E40418E0
470 DATA 0047CFDFFF7F1F0F,07
03,00E2F3FBFFFEF8F0,E0C0
480 DATA 0505050505050505,,4
040404040404040,
490 DATA 0000010204040404,,0
0008040202020202,
500 DATA FFFFFFFFFFE1EFE1,EF
E1EFE1EFE1FFFF,FFFFFFFFFF0FE
F0F,EF0FEF0FEF0FFFFF
510 RESTORE :: FOR I=96 TO 1
23 :: READ A$ :: CALL CHAR(I
,A$):: NEXT I
520 CALL CHAR(142,"010204081
0204080FF")
530 CALL COLOR(12,7,7,10,2,1
6,14,1,16,13,13,13,1,16,16)
540 ! BILDSCHIRMAUFBAU
550 CALL CLEAR :: CALL SCREE
N(2):: CALL MAGNIFY(3)
560 DISPLAY AT(1,3):"0 MIN"&
RPT$(CHR$(129),14)&"PKT   0"
570 FOR I=6 TO 21 STEP 5
580 DISPLAY AT(I,1):"xzxz...
.xzxz....xzxz....xzxzy{y{...
.y{y{....y{y{....y{y{lnln...
.lnln....lnln....lnlnmomo...
.momo....momo....momo"
590 NEXT I
```

```
600 CALL TUERAUF :: CALL SPR
ITE(#3,104,2,17,256,0,-7)::
CALL GONG :: CALL TUERZU
610 DISPLAY AT(1,15-LEN(FACH
$)/2)SIZE(LEN(FACH$)):FACH$
620 CALL POSITION(#3,Y,X)::
IF X<25 THEN 640 ELSE CALL K
EY(0,K,S):: IF S<1 THEN 620
630 IF K=1 THEN 1520 ELSE IF
 K=2 OR K=15 THEN CALL CLEAR
 :: CALL DELSPRITE(ALL):: CA
LL HCHAR(1,1,47,768):: GOTO
1780 ELSE 620
640 CALL MOTION(#3,0,0):: FO
R I=1 TO 300 :: NEXT I
650 ! SPIELBEGINN
660 CALL COLOR(12,7,8)
670 CALL HCHAR(23,19,32,2)::
 CALL HCHAR(24,19,32,2)
680 CALL SPRITE(#1,96,14,177
,145)
690 WERT=INT(RND*5)+1 :: ON
WERT GOSUB 1900,1920,1940,19
60,1980
700 ORTV=INT(RND*4)*64+25 ::
 ORTH=INT(RND*4)*40+42
710 CALL SPRITE(#4,128,16,OR
TH,ORTV)
720 CALL JOYST(1,XJ,YJ)
730 CALL MOTION(#1,-2*YJ,2*X
J)
740 IF XJ=0 AND YJ=0 THEN 76
0
750 Q=-Q :: CALL PATTERN(#1,
98+2*Q)
760 CALL KEY(1,K,S)
770 IF K=18 THEN 840
780 CALL COINC(#1,#4,10,A)::
 IF A=-1 THEN 1130
790 ZEIT=ZEIT+1 :: IF ZEIT/1
0=INT(ZEIT/10)THEN DISPLAY A
T(1,2)SIZE(2):USING "##":ZEI
T/10-.5 :: IF ZEIT>=450 THEN
 1170
800 COUNT=COUNT+1 :: IF COUN
T<50 THEN 720
810 CALL POSITION(#1,Y1,X1,#
3,Y2,X2):: CALL MOTION(#3,0,
SGN(X1-X2)*GES)
820 IF RND<SCHWER THEN 970 E
LSE 720
830 ! SCHUSS MAENNCHEN
840 CALL POSITION(#1,Y1,X1):
: CALL SPRITE(#2,116,13,Y1,X
1,-100,0):: SCH=SCH+1 :: P=M
AX(P-1,0)
850 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
860 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
870 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
880 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
890 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
900 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
910 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
920 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
930 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940 ELSE DISPL
AY AT(1,26):USING "###":P ::
 CALL DELSPRITE(#2):: GOTO 7
80
940 CALL DELSPRITE(#2):: DIS
PLAY AT(1,26):USING "###":P
950 CALL MOTION(#3,0,0):: FO
R I=0 TO 20 STEP 2 :: CALL S
OUND(-100,-5,I):: NEXT I ::
COUNT=0 :: GES=GES+1 :: SCHW
ER=SCHWER+.02 :: GOTO 780
960 ! SCHUSS LEHRER
970 CALL POSITION(#3,Y2,X2):
: CALL SPRITE(#2,112,11,Y2,X
2,100,0)
980 CALL COINC(#1,#2,10,A)::
 IF A=-1 THEN 1090
990 CALL COINC(#1,#2,10,A)::
 IF A=-1 THEN 1090
1000 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1010 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1020 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1030 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1040 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1050 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1060 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1070 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1080 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090 ELSE CAL
L DELSPRITE(#2):: GOTO 720
1090 CALL DELSPRITE(#2):: CA
LL MOTION(#1,0,0,#3,0,0)
1100 FOR I=550 TO 110 STEP -
10 :: CALL SOUND(-100,I,5)::
 NEXT I
1110 P=0 :: SCH=SCH+5 :: GOT
O 1170
1120 ! SPICKPUNKTE GEFRESSEN
1130 CALL MOTION(#1,0,0):: C
ALL DELSPRITE(#4):: CALL SOU
```

```
ND(-100,110,5):: CALL SOUND(
-100,330,5):: CALL SOUND(-10
0,500,5)
1140 P=P+WERT :: DISPLAY AT(
1,26):USING "###":P
1150 COUNT=50 :: GOTO 690
1160 ! STUNDE VORBEI
1170 NOTE(RU)=1
1180 IF P<100 THEN NOTE(RU)=
2
1190 IF P<80 THEN NOTE(RU)=3
1200 IF P<60 THEN NOTE(RU)=4
1210 IF P<40 THEN NOTE(RU)=5
1220 IF P<20 THEN NOTE(RU)=6
1230 CALL DELSPRITE(#4):: CA
LL MOTION(#1,0,0,#3,0,0)
1240 CALL GONG :: CALL COLOR
(12,7,7):: CALL TUERAUF :: C
ALL MOTION(#3,0,7)
1250 CALL POSITION(#3,Y2,X2)
:: IF X2>250 THEN 1280 ELSE
CALL KEY(0,K,S):: IF S=0 THE
N 1250
1260 IF K=1 THEN 1520 ELSE I
F K=2 OR K=15 THEN CALL CLEA
R :: CALL DELSPRITE(ALL):: C
ALL HCHAR(1,1,47,768):: GOTO
 1780
1270 GOTO 1250
1280 CALL DELSPRITE(#3):: CA
LL TUERZU :: CALL HCHAR(1,10
,129,14)
1290 RU=RU+1 :: ON RU GOSUB
1340,1370,1400,1430,1460,149
0,1510
1300 ZEIT,COUNT,P=0 :: CALL
HCHAR(1,27,32,5):: CALL HCHA
R(1,1,32,5)
1310 GOTO 600
1320 ! NEUE STUNDE
1330 ! ERDKUNDE
1340 CALL CHAR(104,"0F1F1804
0F1424380E0F031F3F7F7C30F038
7C7EFE666F57070686C8F")
1350 FACH$="ERDKUNDE" :: GES
=4 :: SCHWER=.08 :: RETURN
1360 ! BIOLOGIE
1370 CALL CHAR(104,"0730405C
422869616240422F3C1B0F07E018
047484282C0C8C0484E878B0E0C0
")
1380 SCHWER=.08 :: FACH$="BI
OLOGIE" :: GES=4 :: RETURN
1390 ! LATEIN
1400 CALL CHAR(104,"00070E10
08101F30508870100C13080700F0
183C3C66D62A0A2A12040408300C0
")
1410 SCHWER=.12 :: FACH$="LA
TEIN" :: GES=5 :: RETURN
1420 ! ENGLISCH
1430 CALL CHAR(104,"030F3C20
1010213F60A0C422211D101F80E0
7070F8F8FCBC1C5C3C1818284080
")
1440 SCHWER=.13 :: FACH$="EN
GLISCH" :: GES=5 :: RETURN
1450 ! DEUTSCH
1460 CALL CHAR(104,"071E3832
30101008080908090A040201F03C
0E2606048488884808C8281020C0
")
1470 SCHWER=.15 :: FACH$="DE
UTSCH" :: GES=6 :: RETURN
1480 ! MATHEMATIK
1490 CALL CHAR(104,"0A0F1F38
32351210101110101309080 7A8F8
FC0E2656240404440404E4C808F0
")
1500 SCHWER=.17 :: FACH$="MA
THEMATIK" :: GES=6 :: RETURN
1510 ! ZEUGNIS
1520 CALL CLEAR :: CALL DELS
PRITE(ALL):: CALL CHARSET ::
 CALL HCHAR(1,1,47,768):: CA
LL TAFEL
1530 CALL SCHRIFT(6,9,"ABSCH
LUSSZEUGNIS")
1540 CALL SCHRIFT(8,9,"FUER
"&NAME$)
1550 CALL SCHRIFT(10,9,"RELI
GION......"&STR$(NOTE(0))):
 CALL SCHRIFT(11,9,"ERDKUNDE
......"&STR$(NOTE(1)))
1560 CALL SCHRIFT(12,9,"BIOL
OGIE......"&STR$(NOTE(2))):
 CALL SCHRIFT(13,9,"LATEIN..
......"&STR$(NOTE(3)))
1570 CALL SCHRIFT(14,9,"ENGL
ISCH......"&STR$(NOTE(4))):
 CALL SCHRIFT(15,9,"DEUTSCH.
......"&STR$(NOTE(5)))
1580 CALL SCHRIFT(16,9,"MATH
EMATIK...."&STR$(NOTE(6)))
1590 IF SCH<26 THEN BETR$="S
EHR GUT"
1600 IF SCH>25 THEN BETR$="G
UT"
1610 IF SCH>50 THEN BETR$="S
CHLECHT"
1620 IF SCH>75 THEN BETR$="S
EHR SCHLECHT"
1630 CALL SCHRIFT(18,9,"BETR
AGEN: "&BETR$)
1640 IF D$<>"J" AND D$<>"j"
THEN 1750
1650 ! ZEUGNIS AUSDRUCKEN
1660 OPEN #1:"RS232"
1670 PRINT #1:TAB(6);"ABSCHL
USSZEUGNIS": : : : :
1680 PRINT #1:TAB(4);"fuer "
```

```
;NAME$: :"Schueler des TI-Gy
mnasiums": : :
1690 PRINT #1:TAB(6);"Religi
on.....";STR$(NOTE(0)): :
TAB(6);"Erdkunde.....";ST
R$(NOTE(1)):
1700 PRINT #1:TAB(6);"Biolog
ie.....";STR$(NOTE(2)): :
TAB(6);"Latein.....";ST
R$(NOTE(3)):
1710 PRINT #1:TAB(6);"Englis
ch.....";STR$(NOTE(4)): :
TAB(6);"Deutsch.....";ST
R$(NOTE(5)):
1720 PRINT #1:TAB(6);"Mathem
atik.....";STR$(NOTE(6)): :
 : :
1730 PRINT #1:"Das Betragen
des Schuelers war durchwegs
";BETR$;".":"Die Mitarbeit l
iess zu wuenschen uebrig."
1740 CLOSE #1
1750 FOR I=1 TO 500 :: CALL
KEY(0,K,S):: IF S<>0 THEN 17
90
1760 NEXT I
1770 ! NEUES SPIEL ?
1780 IF K=15 THEN 1820
1790 CALL TAFEL :: CALL SCHR
IFT(8,3,"WILLST DU WEITERMAC
HEN ODER.....GLEICH VON DER
SCHULE GEHEN?")
1800 CALL KEY(0,K,S):: IF S=
0 THEN 1800
1810 IF K<>74 AND K<>106 THE
N END
1820 CALL TAFEL
1830 CALL SCHRIFT(10,14,"NAM
E:")
1840 ACCEPT AT(13,8)VALIDATE
(UALPHA)SIZE(-14):NAME$
1850 CALL TAFEL :: CALL SCHR
IFT(10,3,"IST DER DRUCKER AN
GESCHLOSSEN")
1860 CALL KEY(0,K,S):: IF S=
0 THEN 1860
1870 D$=CHR$(K):: FOR I=0 TO
 6 :: NOTE(I)=0 :: NEXT I
1880 CALL TAFEL :: GOTO 400
1890 ! CHARDEFINITIONEN FUER
 SPICKPUNKTE
1900 CALL CHAR(128,"00000000
03070F0303030303",130,"",131
,"")
1910 RETURN
1920 CALL CHAR(128,"00000000
0307060000010307070000000000
0000C0E06060E0C080E0E0")
1930 RETURN
1940 CALL CHAR(128,"00000000
0307000303000703000000000000
0000C0E060E0E060E0C0")
1950 RETURN
1960 CALL CHAR(128,"00000000
0606060607070000000000000000
00000000C0C0E0E0C0C0")
1970 RETURN
1980 CALL CHAR(128,"00000000
0707060707000707000000000000
0000E0E000E0E060E0E0")
1990 RETURN
2000 DATA 500,784,500,659,50
0,587,500,659,300,44733
2010 DATA 500,659,500,784,50
0,784,200,698,200,659,500,58
7,300,44733
2020 DATA 400,659,400,622,40
0,659,400,622,400,659,400,62
2,500,784
2030 DATA 200,330,200,311,20
0,330,200,311,200,330,200,31
1,300,349
2040 DATA 200,262,300,262,40
0,294
2050 ! UNTERPROGRAMME
2060 SUB TAFEL
2070 CALL CHAR(47,"FFFFFFFFF
FFFFFFF",128,"FEFEFEFEFEFEFE
FE")
2080 CALL COLOR(13,13,2,9,15
,15,1,13,13,0,13,13):: FOR I
=2 TO 8 :: CALL COLOR(I,16,1
3):: NEXT I
2090 FOR I=19 TO 24 :: CALL
HCHAR(I,11,96,12):: NEXT I
2100 FOR I=1 TO 7 :: CALL VC
HAR(6,I,32,13):: NEXT I
2110 CALL VCHAR(6,8,128,13)
2120 FOR I=9 TO 23 :: CALL V
CHAR(6,I,32,13):: NEXT I
2130 CALL VCHAR(6,24,128,13)
2140 FOR I=25 TO 32 :: CALL
VCHAR(6,I,32,13):: NEXT I
2150 SUBEND
2160 SUB TITEL(D$,NAME$)
2170 CALL CLEAR :: CALL SCRE
EN(2):: CALL HCHAR(1,1,47,76
8):: CALL TAFEL
2180 CALL SCHRIFT(8,11,"WILL
KOMMEN ZU"):: CALL VCHAR(12,
12,47,5):: CALL HCHAR(12,13,
47,3)
2190 CALL HCHAR(16,13,47,3):
: CALL HCHAR(14,13,47)
2200 FOR I=17 TO 21 :: CALL
HCHAR(I-5,I,47):: CALL HCHAR
(33-I,I,47):: NEXT I
2210 CALL MUSIK :: CALL TAFE
L
2220 CALL SCHRIFT(10,14,"NAM
E:"):: ACCEPT AT(13,8)VALIDA
```

```
TE(UALPHA)SIZE(-14):NAME$
2230 CALL TAFEL
2240 CALL SCHRIFT(10,3,"IST
DER DRUCKER ANGESCHLOSSEN")
2250 CALL KEY(0,K,S):: IF S=
0 THEN 2250
2260 D$=CHR$(K)

2270 CALL TAFEL :: CALL SCHR
IFT(10,3,"WILLST DU DIE SPIE
LANLEITUNG")

2280 CALL KEY(0,K,S):: IF S=
0 THEN 2280
2290 IF K<>74 AND K<>106 THE
N CALL TAFEL :: GOTO 2320
2300 CALL TAFEL :: CALL SCHR
IFT(7,2,"OBWOHL DU WEISST,DA
SS IHR        HEUTE EINIGE EX
EN SCHREIBEN      WERDET,HAST
 DU NICHTS GELERNT.")
2310 CALL SCHRIFT(13,2,"SAMM
LE SPICKPUNKTE UND LEGE.....
LEHRER MIT UHAKEN LAHM. ....
....HUETE DICH VOR SEHSTRAHL
EN.")
2320 SUBEND
2330 SUB GONG
2340 FOR I=550 TO 220 STEP -
110 :: FOR J=0 TO 18 :: CALL
 SOUND(-100,I,J):: NEXT J ::
 NEXT I
2350 SUBEND
2360 SUB TUERAUF
2370 FOR I=1 TO 3 :: CALL HC
HAR(5-I,29+I,142):: NEXT I
2380 FOR I=1 TO 100 :: NEXT
I
2390 FOR I=1 TO 3 :: CALL HC
HAR(5-I,30,32,3):: NEXT I
2400 CALL HCHAR(2,29,143,4)
2410 SUBEND
2420 SUB TUERZU
2430 CALL HCHAR(2,29,32,4)
2440 FOR I=1 TO 3 :: CALL HC
HAR(5-I,29+I,142):: NEXT I
2450 FOR I=1 TO 100 :: NEXT
I
2460 FOR I=1 TO 3 :: CALL HC
HAR(5-I,29+I,32):: NEXT I
2470 SUBEND
2480 SUB SCHRIFT(Z,S,A$)
2490 FOR I=1 TO LEN(A$)
2500 IF S-1+I>32 THEN Z=Z+2
:: S=S-32
2510 CALL HCHAR(Z,S-1+I,ASC(
SEG$(A$,I,1)))
2520 FOR J=1 TO 10 :: NEXT J
 :: NEXT I
2530 SUBEND
2540 SUB MUSIK
2550 FOR J=1 TO 2 :: RESTORE
 2610 :: FOR I=1 TO 23 :: RE
AD A,B,C,D :: CALL SOUND(A,B
,0,C,9,D,9)
2560 CALL KEY(0,K,S):: IF S<
>0 THEN 2700
2570 NEXT I :: NEXT J
2580 FOR J=1 TO 2 :: RESTORE
 2660 :: FOR I=1 TO 20 :: RE
AD A,B,C,D :: CALL SOUND(A,B
,0,C,9,D,9)

2590 CALL KEY(0,K,S):: IF S<
>0 THEN 2700
2600 NEXT I :: NEXT J :: GOT
O 2550
2610 DATA 300,494,494,494,10
0,523,523,523,400,587,147,14
7,400,587,185,220,300,587,18
5,220,100,494,185,220
2620 DATA 300,440,262,165,10
0,494,262,165,400,392,196,19
6,400,392,294,247,400,392,29
4,247
2630 DATA 300,494,494,494,10
0,494,494,494,300,440,220,22
0,100,392,220,220,400,330,16
5,165,300,330,262,220
2640 DATA 100,330,262,220,30
0,494,196,196,100,494,196,19
6,400,440,185,185,400,440,22
0,220,800,440,262,262
2650 ! DATA 400,44733,44733,
44733
2660 DATA 400,587,247,196,40
0,587,247,196,400,587,247,19
6,300,587,247,196,100,659,24
7,196
2670 DATA 300,440,220,185,10
0,440,220,185,300,440,220,18
5,100,440,220,185,800,440,22
0,185
2680 DATA 300,440,220,185,10
0,440,220,185,300,440,220,18
5,100,440,220,185,800,440,22
0,185
2690 DATA 300,392,247,196,10
0,392,247,196,300,392,247,19
6,100,392,247,196,800,392,24
7,196

2700 SUBEND

2710 !
2720 ! ********************
2730 !
2740 !  AEHNLICHKEITEN DER
2750 !  LEHRER MIT LEBEN-
2760 !  DEN ODER TOTEN PER-
2770 !  SONEN SIND REIN ZU-
2780 !       FAELLIG !!
2790 !
2800 ! ********************
```

```
10 ! ************************
11 ! *                      *
12 ! *     ADVANTAGE        *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *  Joerg Ruebesamen    *
16 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     4025 Bytes       *
28 ! *                      *
29 ! ************************
120 CALL CLEAR
130 FOR I=1 TO 72 :: PRINT "
ADVANTAGE";
140 NEXT I
150 DISPLAY AT(10,1):"......
.............................R
UEBIOLA 1984 LTD............
........................."
160 OPTION BASE 1
170 FOR I=1 TO 7 :: CALL COL
OR(I,16,15):: NEXT I :: CALL
 COLOR(10,10,15,11,16,15,12,
2,15,9,13,15,8,16,15,0,15,15
)
180 DIM AM(3),K(4,30,2),C(3)
190 C(1)=120 :: C(2)=104 ::
C(3)=112
200 IF RND>.5 THEN PL=1 ELSE
 PL=1.5
210 RANDOMIZE
220 ON WARNING NEXT
230 DATA C0FFFFFF1C0E0703,03
01,03FFFFFF3870E0C0,C0E0F078
3C1E0F07,0000000000000008
240 DATA 0000000000000001,03
070F1E3C78F0E,C08,1818181818
181818
250 DATA 4,4,4,10,4,16
260 DATA 9,1,9,7,9,13,9,19
270 DATA 14,4,14,10,14,16
280 DATA 19,1,19,7,19,13,19,
19
290 A$="0000000A041F040A" ::
 CALL CHAR(120,A$,104,"00000
01F1F1F1F1F",112,A$)
300 FOR I=88 TO 96 :: READ A
$ :: CALL CHAR(I,A$):: NEXT
I
310 CALL HCHAR(1,1,30,768)
320 FOR I=6 TO 21 STEP 3 ::
CALL HCHAR(1,I,49+A):: A=A+1
 :: NEXT I
330 FOR I=1 TO 27 STEP 3 ::
CALL VCHAR(1,I,96,24):: NEXT
 I
340 FOR I=1 TO 14 :: READ A,
B :: IF RND<.5 THEN CALL SPL
(A,B)ELSE CALL SPR(A,B)
350 NEXT I
360 RESTORE 250 :: FOR A=1 T
O 14 :: READ I,J
370 DISPLAY AT(I-1,J+1)SIZE(
2):CHR$(30)&CHR$(30):: NEXT
A
380 I=2 :: J=9
390 PL=PL+.5 :: DISPLAY AT(2
2,24):">>"&STR$(INT(INT(PL)/
PL)+1)&"<<"
400 DISPLAY AT(1,24):"SP.:1"
 :: DISPLAY AT(11,24):"SP.:2
"
410 B=4 :: CALL PUNKT(B,A,PL
,K(,,),C())
420 DISPLAY AT(24,1):" x=1
 h=2   p=3  " :: ACCEPT AT(2
4,20)SIZE(1)VALIDATE("123")B
EEP:F :: DISPLAY AT(24,1):"W
ELCHE SPALTE ?"
430 ACCEPT AT(24,20)SIZE(1)V
ALIDATE("123456")BEEP:SP
440 J=SP*3+2 :: I=2
450 FOR A=1 TO 30 :: IF K(F,
A,1)=0 THEN 470
460 NEXT A
470 IF A>AM(F)THEN AM(F)=A
480 K(F,A,1)=I :: K(F,A,2)=J
 :: CALL HCHAR(K(F,A,1),K(F,
A,2),C(F))
490 CALL ROLL(K(,,),C(),F,A,
PL):: IF @=0 THEN 500 ELSE 4
90
500 CALL SOUND(10,1000,10)::
 FOR T=1 TO 3 :: FOR B=1 TO
3
510 FOR A=1 TO AM(B):: IF K(
B,A,2)>0 THEN CALL ROLL(K(,,
),C(),B,A,PL)
520 NEXT A
530 NEXT B
540 NEXT T
550 GOTO 390
560 SUB SPL(I,J)
570 CALL SOUND(100,-6,20)
580 DISPLAY AT(I,J)SIZE(3):"
XZ"&CHR$(30):: DISPLAY AT(I+
1,J)SIZE(3):"Y[\" :: DISPLAY
 AT(I+2,J)SIZE(3):CHR$(30)&"
Y["
590 SUBEND
600 SUB SPR(I,J)
610 CALL SOUND(100,-5,20)
620 DISPLAY AT(I,J)SIZE(3):C
HR$(30)&"XZ" :: DISPLAY AT(I
+1,J)SIZE(4):"]^_"&CHR$(30):
: DISPLAY AT(I+2,J)SIZE(4):"
^_"&CHR$(30)&CHR$(30)
```

```
630 I=I+2
640 SUBEND
650 SUB ROLL(K(,,),C(),B,A,P
L)
660 I=K(B,A,1):: J=K(B,A,2):
: GOTO 680
670 K(B,A,1)=I :: K(B,A,2)=J
680 CALL GCHAR(I+1,J,G):: CA
LL GCHAR(I+1,J+1,G1)
690 IF G<>120 AND G<>104 AND
 G<>112 THEN 720 ELSE IF I<>
2 AND I<>7 AND I<>12 AND I<>
17 THEN 780
700 CALL GCHAR(I+1,J+2,G)::
CALL GCHAR(I+1,J-1,G1):: IF
G1<>30 AND G<>30 THEN 780 EL
SE CALL HCHAR(I,J,30):: I=I+
2 :: IF G1=96 THEN J=J+3 ELS
E IF G=96 THEN J=J-3
710 CALL HCHAR(I,J,C(B)):: C
ALL SOUND(20,-2,0):: GOTO 67
0
720 IF G+G1<>60 THEN 750
730 CALL HCHAR(I,J,30):: I=I
+1 :: IF I>22 THEN CALL VCHA
R(23,K(B,A,2),30,1):: K(B,A,
1),K(B,A,2)=0 :: CALL PUNKT(
B,A,PL,K(,,),C()):: GOTO 780
 ELSE CALL HCHAR(I,J,C(B))
740 GOTO 680
750 IF G>30 AND G<91 OR G1>3
0 AND G1<91 THEN 770
760 IF G1>30 THEN CALL SPL(I
,J-1):: GOTO 730 ELSE IF G>3
0 THEN CALL SPR(I,J-4):: GOT
O 730
770 K(B,A,1)=I :: K(B,A,2)=J
 :: @=0
780 SUBEND
790 SUB PUNKT(B,A,PL,K(,,),C
())
800 DIM FI(4,4,2),FE(4,4,2)
810 IF BRETT=0 THEN CALL BR(
FE(,,),FI(,,),C()):: BRETT=8
820 K(B,A,2),K(B,A,1)=0
830 IF PL=INT(PL)THEN 890
840 SUM,Z=0
850 FOR I=1 TO 4 :: FOR J=1
TO 4 :: SUM=SUM+FE(I,J,2)::
IF FE(I,J,1)=B AND FE(I,J,2)
=1 AND Z=0 THEN CALL HCHAR(I
+2,J+26,30):: FE(I,J,2)=7 ::
 Z=2 :: SUM=SUM+6
860 NEXT J :: NEXT I
870 IF SUM>=16*7 THEN CALL E
NDE(1)
880 SUBEXIT
890 SUM,Z=0
900 FOR I=1 TO 4 :: FOR J=1
TO 4 :: SUM=SUM+FI(I,J,2)::
IF FI(I,J,1)=B AND FI(I,J,2)
=1 AND Z=0 THEN CALL HCHAR(I
+12,J+26,30):: FI(I,J,2)=7 :
: Z=2 :: SUM=SUM+6
910 NEXT J :: NEXT I
920 IF SUM=16*7 THEN CALL EN
DE(2)
930 SUBEND
940 SUB BR(FE(,,),FI(,,),C()
)
950 FOR I=1 TO 4 :: FOR J=1
TO 4
960 IF RND<.4 THEN FI(I,J,1)
=1 ELSE IF RND<.65 THEN FI(I
,J,1)=2 ELSE FI(I,J,1)=3
970 CALL HCHAR(I+12,J+26,C(F
I(I,J,1)))
980 IF RND<.4 THEN FE(I,J,1)
=1 ELSE IF RND<.65 THEN FE(I
,J,1)=2 ELSE FE(I,J,1)=3
990 CALL HCHAR(I+2,J+26,C(FE
(I,J,1)))
1000 FE(I,J,2),FI(I,J,2)=1
1010 NEXT J :: NEXT I
1020 SUBEND
1030 SUB ENDE(I)
1040 DISPLAY AT(10,1)BEEP ER
ASE ALL:"SPIELER";I;"HAT GEW
ONNEN"
1050 DISPLAY AT(14,1):"1..NO
CHMAL.....................
.....................2..SC
HLUSS"
1060 CALL KEY(0,K,S):: IF K=
49 THEN RUN ELSE IF K=50 THE
N STOP ELSE CALL SOUND(4201,
262,16,370,17,311,17):: GOTO
 1060
1070 SUBEND
```

BÖRSE

Verkaufe Module wie Topper/Princess/Driving/ einmalig in der BRD + ExBasicspiele ab 5 DM / Module ab 20 DM. Liste gegen DM 1 bei Wolfgan Auer, Kantstr. 18, 645 Hanau

---

Verk. TI-99/4A + ExBasic und Lerncass. + Cass. Rec. + Rec. Kabel + Soccer + Progr. + 2 Bücher + Infos + Joyst. VB eventuell auch einzeln. Tele 07135/5744 und Atari 2600 mit 3 Modulen und Joyst. Arpad Huber Jr., 7129 Haberschlacht

---

Verk. P-Box mit V 24 und Cent., Disklw, Ds-Controller, 64K-Bausatz, kaum gebraucht, mit Software. VB 1700. Tel. 05323/1811 n. 18 Uhr.

TI99/4A + EXT-Basic + Schach + Kass-Rec-Kabel + Alien-Add + Minus-Mis + Software + Lit; VB 520,–. Tel. 0201/30 67 48 ab 17.30 Uhr

---

Verkaufe TI-99/4A + Ext.-Basic + 150 Pgm. sowie Module. Info: H. Hollmann, Tel. 0209/84144

```
10 ! ********************
11 ! *                  *
12 ! *  STEUERBERECHNUNG  *
13 ! *                  *
14 ! *    Copyright by  *
15 ! *                  *
16 ! *   Thomas Ehlert  *
17 ! *                  *
19 ! * Benoetigte Gerate *
20 ! *  TI99/4A  Konsole *
21 ! *     Ext. Basic   *
22 ! *   opt.: Drucker  *
23 ! *                  *
26 ! * Speicherbelegung *
27 ! *      6902 Bytes  *
28 ! *                  *
29 ! ********************
30 !
280 CALL CLEAR
290 CALL SCREEN(12)
300 DISPLAY AT(1,6):"STEUERB
ERECHNUNG"
310 DISPLAY AT(2,6):"-------
---------"
320 DISPLAY AT(3,6):"@ T.Ehl
ert  3/85 "
330 DISPLAY AT(5,1):"DAS PRO
GRAMM BERECHNET JAH-"
340 DISPLAY AT(6,1):"RESSTEU
ER,KIRCHENSTEUER UND"
350 DISPLAY AT(7,1):"MONATLI
CHES NETTOEINKOMMEN"
360 DISPLAY AT(8,1):"UNTER B
ERUECKSICHTIGUNG EI-"
370 DISPLAY AT(9,1):"NER ETW
AIGEN STEUERERSPARNIS"
380 DISPLAY AT(10,1):"NACH P
ARAGRAPH 7b EStG, SO-"
390 DISPLAY AT(11,1):"WIE SA
EMTLICHE SOZIALVERSI-"
400 DISPLAY AT(12,1):"CHERUN
GSBEITRAEGE VON SOZI-"
410 DISPLAY AT(13,1):"ALVERS
ICHERUNGSPFLICHTIGEN"
420 DISPLAY AT(14,1):"ARBEIT
NEHMERN"
430 DISPLAY AT(20,7):"EINE T
ASTE DRUECKEN!"
440 CALL KEY(0,K,S):: IF S=0
 THEN 440
450 CALL CLEAR
460 DISPLAY AT(1,1):""
470 DISPLAY AT(2,1):"ERKLAER
UNG DER EINGABEWERTE:"
480 DISPLAY AT(3,1):"-------
-----------------------"
490 DISPLAY AT(5,1):"~BRUTTO
JAHRESEINKOMMEN BEIN-HALTET
SAEMTLICHE STEUER-...PFLICHT
IGEN EINKUENFTE, ALSO AUCH U
RLAUBS- UND WEIHNACHTSGELD"
500 DISPLAY AT(9,1):"SOWIE S
TEUERPFLICHTIGE SPAR-ZULAGEN
 ETC. ..............~WERBUN
GSKOSTEN SIND IN TAT-SAECHLI
CHER HOEHE, HILFSWEI-"
510 DISPLAY AT(13,1):"SE ALS
 PAUSCHALBETRAG EINZU-GEBEN.
"
520 DISPLAY AT(15,1):"~SONDE
RAUSGABEN SIND IN TAT-SAECHL
ICHER HOEHE, HILFSWEI-"
530 DISPLAY AT(17,1):"SE ALS
 PAUSCHALBETRAG EINZU-GEBEN.
"
540 DISPLAY AT(19,1):"~HAUS-
BAUKOSTEN BEINHALTEN  NUR DI
E HERSTELLUNGS- NICHT ABER D
IE GRUNDSTUECKSKOSTEN"
550 DISPLAY AT(22,7):"EINE T
ASTE DRUECKEN!"
560 CALL KEY(0,K,S):: IF S=0
 THEN 560
570 CALL CLEAR
580 DISPLAY AT(2,1):"IM FOLG
ENDEN ERHALTEN SIE...DIE ZUR
 HILFSWEISEN EINGABE ERFORDE
RLICHEN"
590 DISPLAY AT(6,6):"PAUSCHA
LBETRAEGE"
600 DISPLAY AT(7,6):"-------
---------"
610 DISPLAY AT(9,1):"FUER WE
RBUNGSKOSTEN UND SON-DERAUSG
ABEN:"
620 DISPLAY AT(12,1):"ABKUER
ZUNGSERKLAERUNG:"
630 DISPLAY AT(13,1):"LED=LE
DIG VH=VERHEIRATET....SVP=SO
ZIALVERS.PFLICHTIG....DOP=DO
PPELVERDIENER"
640 DISPLAY AT(17,1):"SIND S
IE (1) LED + SVP.....
...(2) LED + NICHT SVP.....
...(3) VH  + SVP.....
...(4) VH  + NICHT SVP.....
...(5) DOP + SVP"
650 DISPLAY AT(22,1):".....
...(6) DOP + NICHT SVP.....
.....................BITTE
KENNZAHL EINGEBEN!"
660 ACCEPT AT(24,26)BEEP VAL
IDATE("123456")SIZE(1):Q
670 IF Q=1 THEN 730
680 IF Q=2 THEN 750
690 IF Q=3 THEN 770
700 IF Q=4 THEN 790
710 IF Q=5 THEN 810
720 IF Q=6 THEN 830
730 CALL CLEAR :: DISPLAY AT
(5,1):"WERBUNGSKOSTENPAUSCHA
```

```
LE  564"
740 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE 3780" :: GO
TO 850
750 CALL CLEAR :: DISPLAY AT
(5,1):"WERBUNGSKOSTENPAUSCHA
LE  564"
760 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE 2268" :: GO
TO 850
770 CALL CLEAR :: DISPLAY AT
(5,1):"WERBUNGSKOSTENPAUSCHA
LE  564"
780 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE......OHNE K
INDER....7560.........EIN  K
IND......8424.........ZWEI K
INDER....8656.........DREI K
INDER....9130" :: GOTO 850
790 CALL CLEAR :: DISPLAY AT
(5,1):"WERBUNGSKOSTENPAUSCHA
LE  564"
800 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE......OHNE K
INDER....4536.........EIN  K
IND......5400.........ZWEI K
INDER....6318.........DREI K
INDER....7236" :: GOTO 850
810 CALL CLEAR :: FB=2 :: DI
SPLAY AT(5,1):'WERBUNGSKOSTE
NPAUSCHALE 1128"
820 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE......OHNE K
INDER....7560.........EIN  K
IND......8424.........ZWEI K
INDER....9342.........DREI K
INDER...10260" :: GOTO 850
830 CALL CLEAR :: FB=2 :: DI
SPLAY AT(5,1):"WERBUNGSKOSTE
NPAUSCHALE 1128"
840 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE......OHNE K
INDER....4536.........EIN  K
IND......5400.........ZWEI K
INDER....6318.........DREI K
INDER....7236" :: GOTO 850
850 DISPLAY AT(18,1):"BITTE
MERKEN SIE SICH DIE...FUER S
IE RELEVANTEN BETRAEGEUND ST
ARTEN SIE DAS PROGRAMMMIT EI
NEM DEZENTEN DRUCK AUFIRGEND
 EINE TASTE!"
860 CALL KEY(0,K,S):: IF S=0
 THEN 860
870 CALL CLEAR
880 FB=1
890 DISPLAY AT(5,6):"BITTE G
EBEN SIE DIE"
900 DISPLAY AT(7,6):"ARBEITN
EHMERANTEILE"
910 DISPLAY AT(9,6):"IN % FU
ER FOLGENDE"
920 DISPLAY AT(11,6):"VERSIC
HERUNGEN EIN:"
930 DISPLAY AT(17,1):"KRANKE
NVERSICHERUNG?"
940 ACCEPT AT(17,22)BEEP VAL
IDATE(NUMERIC)SIZE(5):KV
950 DISPLAY AT(19,1):"RENTEN
VERSICHERUNG ?"
960 ACCEPT AT(19,22)BEEP VAL
IDATE(NUMERIC)SIZE(5):RV
970 DISPLAY AT(21,1):"ARBEIT
SLOSENVERS.  ?"
980 ACCEPT AT(21,22)BEEP VAL
IDATE(NUMERIC)SIZE(5):ALV
990 CALL CLEAR
1000 DISPLAY AT(2,1):"BRUTTO
JAHRESEINK.? "
1010 ACCEPT AT(2,21)BEEP VAL
IDATE(NUMERIC)SIZE(9):BJ
1020 DISPLAY AT(3,1):"WERBUN
GSKOSTEN...? "
1030 ACCEPT AT(3,21)BEEP VAL
IDATE(NUMERIC)SIZE(9):WK
1040 DISPLAY AT(4,1):"SONDER
AUSGABEN...? "
1050 ACCEPT AT(4,21)BEEP VAL
IDATE(NUMERIC)SIZE(9):SA
1060 DISPLAY AT(5,1):"HAUS-B
AUKOSTEN...? "
1070 ACCEPT AT(5,21)BEEP VAL
IDATE(NUMERIC)SIZE(9):HBK
1080 DISPLAY AT(6,1):"ZAHL D
ER KINDER  ? "
1090 ACCEPT AT(6,21)BEEP VAL
IDATE(NUMERIC)SIZE(2):KZ
1100 DISPLAY AT(7,1):"SPLITT
ING(J=2/N=1) "
1110 ACCEPT AT(7,21)BEEP VAL
IDATE(NUMERIC)SIZE(1):SP
1120 IF HBK>250000 THEN HBK=
250000
1130 STSP=(HBK/100)*5
1140 JG=BJ-SA-WK-(1080*FB)-(
KZ*432)-STSP
1150 A=JG
1160 ON SP GOTO 1170,1190
1170 V=INT(JG/54)*54
1180 GOTO 1200
1190 V=(INT(JG/108)*108)/SP
1200 IF V<4213 THEN 1250
1210 IF V<18000 THEN 1260
1220 IF V<60000 THEN 1280
1230 IF V<130000 THEN 1310
1240 IF V>129999 THEN 1340
1250 DISPLAY AT(15,5):"KEINE
 BERECHNUNG" :: GOTO 1890
1260 F=.22*V-926
1270 GOTO 1350
1280 H=(V-18000)/10000
```

```
1290 F=(((3.05*H-73.76)*H+69
5)*H+2200)*H+3034
1300 GOTO 1350
1310 M=(V-60000)/10000
1320 F=(((.09*M-5.45)*M+88.1
3)*M+5040)*M+20018
1330 GOTO 1350
1340 F=.56*V-14837
1350 IF SP=2 THEN F=F*2
1360 DISPLAY AT(9,1):"ZU VER
ST. EINK.:" :: DISPLAY AT(9,
20):INT(JG)
1370 DISPLAY AT(11,1):"EINKO
MMENSTEUER:" :: DISPLAY AT(1
1,20):INT(F)
1380 IF KZ=1 THEN Z=600 :: G
OTO 1440
1390 IF KZ=2 THEN Z=1560 ::
GOTO 1440
1400 IF KZ=3 THEN Z=3360 ::
GOTO 1440
1410 IF KZ=4 THEN Z=5160 ::
GOTO 1440
1420 IF KZ=5 THEN Z=6960 ::
GOTO 1440
1430 IF KZ=6 THEN Z=8760
1440 Y=F-Z
1450 X=.09*Y :: IF X<0 THEN
X=0
1460 DISPLAY AT(13,1):"KIRCH
ENSTEUER  :" :: DISPLAY AT(1
3,20):INT(X)
1470 AKV=((BJ/100)*KV)/13
1480 ARV=((BJ/100)*RV)/13
1490 AALV=((BJ/100)*ALV)/13
1500 SVGES=AKV+ARV+AALV
1510 DISPLAY AT(20,1):"SIND
SIE SOZIALVERSICHERUNGSPFLIC
HTIG.....
.....(J/N)"
1520 ACCEPT AT(22,28)BEEP VA
LIDATE("JN")SIZE(1):A$
1530 IF A$="J" THEN 1550
1540 IF A$="N" THEN 1780
1550 DISPLAY AT(20,1):""
1560 DISPLAY AT(21,1):""
1570 DISPLAY AT(15,1):"KRANK
ENVERS.BEITRAG:"
1580 DISPLAY AT(15,20):INT(A
KV)
1590 DISPLAY AT(17,1):"RENTE
NVERS.BEITRAG :"
1600 DISPLAY AT(17,20):INT(A
RV)
1610 DISPLAY AT(19,1):"ARBEI
TSLOSENVERS.  :"
1620 DISPLAY AT(19,20):INT(A
ALV)
1630 NG=BJ-F-X-(SVGES*13)
1640 MG=NG/13
1650 DISPLAY AT(21,1):"MONAT
LICH NETTO:"
1660 DISPLAY AT(21,20):INT(M
G)
1670 DISPLAY AT(22,1):"-----
----------------------"
1680 DISPLAY AT(24,1):"WERTE
 AUSDRUCKEN....(J/N)"
1690 DISPLAY AT(15,25):")" :
: DISPLAY AT(16,25):")" :: D
ISPLAY AT(17,25):")MTL"
1700 DISPLAY AT(18,25):")" :
: DISPLAY AT(19,25):")"
1710 ACCEPT AT(24,28)BEEP VA
LIDATE("JN")SIZE(1):A$
1720 IF A$="J" THEN 1930
1730 IF A$="N" THEN 1740
1740 DISPLAY AT(24,1):"NEUE
BERECHNUNG.....(J/N)"
1750 ACCEPT AT(24,28)BEEP VA
LIDATE("JN")SIZE(1):A$
1760 IF A$="J" THEN CALL CLE
AR :: GOTO 570
1770 IF A$="N" THEN CALL CLE
AR :: STOP
1780 DISPLAY AT(20,1):""
1790 DISPLAY AT(21,1):""
1800 NG=BJ-F-X
1810 MG=NG/13
1820 DISPLAY AT(21,1):"MONAT
LICH NETTO:"
1830 DISPLAY AT(21,20):INT(M
G)
1840 DISPLAY AT(22,1):"-----
----------------------"
1850 DISPLAY AT(24,1):"WERTE
 AUSDRUCKEN  (J/N)"
1860 ACCEPT AT(24,26)BEEP VA
LIDATE("JN")SIZE(1):A$
1870 IF A$="J" THEN 1930
1880 IF A$="N" THEN 1890
1890 DISPLAY AT(24,1):"NEUE
BERECHNUNG...(J/N)"
1900 ACCEPT AT(24,26)BEEP VA
LIDATE("JN")SIZE(1):A$
1910 IF A$="J" THEN CALL CLE
AR :: GOTO 890
1920 IF A$="N" THEN CALL CLE
AR :: STOP
1930 OPEN #1:"PIO"
1940 FOR Z=1 TO 23
1950 FOR P=1 TO 30
1960 CALL GCHAR(Z,P,X)
1970 DRU$=SEG$(DRU$,1,P)&CHR
$(X)
1980 NEXT P
1990 PRINT #1:DRU$
2000 NEXT Z
2010 CLOSE #1
2020 GOTO 1740
```

```
10 REM*********************
11 REM*                   *
12 REM*  KOMPOSITIONEN    *
13 REM*  Copyright  by    *
14 REM*                   *
15 REM* A.Saberschinsky   *
16 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*  Kass.-Rekorder   *
20 REM*                   *
23 REM* Speicherbelegung *
24 REM*    9172 Bytes     *
25 REM*********************
26 REM
100 DIM F1(125),F2(125),F3(1
25),L(125)
110 CALL CLEAR
120 PRINT "*NOTENKOMPOSITION
SPROGRAMM*": : : : : : : : :
 : :"ENTWICKELT UND GESCHRIE
BEN  VON"
130 PRINT ".....ALEXANDER SA
BERSCHINSKY1985": : : :
140 CALL SOUND(100,110,0,100
0,0,10000,0)
150 FOR Z=1 TO 2500
160 NEXT Z
170 FLAG1=0
180 FLAG11=0
190 FLAG3=0
200 CALL CLEAR
210 PRINT ".........WAHLLIST
E.........================
==========="
220 PRINT "................
.........BEDIENUNGSANLEITU
NG........1NOTENEINGABE.....
.........2"
230 PRINT "NOTENKORREKTUR...
.........3ABSPIELEN DER KOM
POSITION..4LADEN EINER KOMPO
SITION....5"
240 PRINT "SPEICHERN EINER K
OMPOSITON.6ENDE............
.........7": : : : : :
250 CALL KEY(0,K,S)
260 IF S=0 THEN 250
270 IF K<49 THEN 250
280 IF K>55 THEN 250
290 ON K-48 GOTO 300,690,115
0,1540,1610,1690,1770
300 CALL CLEAR
310 PRINT "MIT DIESEM PROGRA
MM KOENNEN SIE BELIEBIGE MEL
ODIEN EIN- GEBEN, DIE NACH E
INER EVEN-"
320 PRINT "TUELLEN KORREKTUR
 ANGEHOERT ODER GESPEICHERT
WERDEN.....KOENNEN. AUCH DAS
 LADEN "
330 PRINT "ALTER MELODIEN VO
N CASSETTE IST MOEGLICH."
340 PRINT "DEN TONWERT GEBEN
 SIE EIN,..INDEM SIE FUER DI
E ENT-.....SPRECHENDE NOTE D
IE TASTE.."
350 PRINT "AUF DER ALPHANUMM
ERISCHEN...TASTATUR DRUECKEN
. FUER.....DIE MITTLERE TONL
EITER IST"
360 PRINT "NUR DIE ENTSPRECH
ENDE TASTE ZU DRUECKEN, FUER
 DIE TIEFE TONLEITER ZUSAETZ
LICH DIE "
370 PRINT "TASTE 'FCTN' UND
FUER DIE...HOHE TONLEITER DI
E TASTE....'CTRL'. (Z.B.: HO
HES G -->..CTRL G.)": :
380 PRINT ">BITTE EINE TASTE
 DRUECKEN<"
390 CALL KEY(0,K,S)
400 IF S=0 THEN 390
410 CALL CLEAR
420 PRINT "BITTE MERKEN SIE
SICH.......WEITERHIN FOLGEND
E ZU-.......SAETZLICHE TASTEN
BELEGUNG...(EV. NOTIEREN):"
430 PRINT "..NOTE......!...T
ASTE.......================
==========..C#, D&....!....
..5......."
440 PRINT "..D#, E&....!....
..6.........F#, G&....!....
..7.........G#, A&....!....
..8......."
450 PRINT "..A#, B&....!....
..9.......----------------
----------": :
460 PRINT "# = ERHOEHUNG EIN
ER NOTE....& = VERMINDERUNG
EINER NOTE": :
470 PRINT "FUER DIE TONLEITE
RWAHL BITTEWIEDER 'FCTN' BZW
. 'CTRL'...VERWENDEN.": : :
480 PRINT ">BITTE EINE TASTE
 DRUECKEN<"
490 CALL KEY(0,K,S)
500 IF S=0 THEN 490
510 CALL CLEAR
520 PRINT "DIE TASTEN FUER D
IE NOTEN-..LAENGE SIND WIE F
OLGT BE-...LEGT:"
530 PRINT "NOTENLAENGE !...T
ASTE.......================
==========..32STEL....!....
..1......."
540 PRINT "..16TEL.....!....
..2...........8TEL.....!....
..3...........4TEL.....!....
..4......."
```

```
550 PRINT "...4TEL . ..!....
..5...........HALBE....!....
..6...........GANZE....!....
..7......."
560 PRINT "-----------------
----------JEWEILS MIT 'SHIF
T' !!!!!!!!----------------
----------": : :
570 PRINT ">BITTE EINE TASTE
 DRUECKEN<"
580 CALL KEY(0,K,S)
590 IF S=0 THEN 580
600 CALL CLEAR
610 PRINT "FUER EINE PAUSE D
RUECKEN DIEBITTE BEI DER NOT
ENHOEHEN-..WAHL EIN 'P' UND
DANN DIE"
620 PRINT "GEWUENSCHTE LAENG
E. ........BEI DER NOTENEING
ABE KOENNENSIE ZWISCHEN EIN-
, ZWEI-,"
630 PRINT "UND DREIKLANG (OD
ER AUCH BE-ENDIGUNG DER EING
ABEN) WAEH-LEN."
640 PRINT "WENN SIE WIEDERHO
LT (2).....AUS DER WAHLLISTE
 WAEHLEN,..WIRD DIE GESPEICH
ERTE KOMPO-"
650 PRINT "SITION GELOESCHT!
!!": : : : :">BITTE EINE TAS
TE DRUECKEN<"
660 CALL KEY(0,K,S)
670 IF S=0 THEN 660
680 GOTO 170
690 CALL CLEAR
700 A=0
710 A=A+1
720 PRINT "NOTENEINGABE:....
.....^^^^^^^^^^^^^^":"-
>(1) EINKLANG.............-
>(2) ZWEIKLANG............-
>(3) DREIKLANG............-
>(4) ENDE DER EINGABE": :
730 CALL KEY(0,K,S)
740 IF S=0 THEN 730
750 IF K<49 THEN 730
760 IF K>52 THEN 730
770 ON K-48 GOTO 780,850,960
,1110
780 FLAG4=1
790 GOSUB 1920
800 CALL SOUND(L(A),F1(A),0)
810 F2(A)=F1(A)
820 F3(A)=F1(A)
830 IF FLAG3=1 THEN 170
840 GOTO 710
850 FLAG2=1
860 FLAG4=2
870 PRINT "-------- 1. KLANG
-----------"
880 GOSUB 1920
890 FLAG5=1
900 PRINT "-------- 2. KLANG
-----------"
910 GOSUB 1920
920 CALL SOUND(L(A),F1(A),0,
F2(A),0)
930 F3(A)=F2(A)
940 IF FLAG3=1 THEN 170
950 GOTO 710
960 FLAG2=1
970 FLAG4=3
980 FLAG6=1
990 PRINT "-------- 1. KLANG
-----------"
1000 GOSUB 1920
1010 FLAG2=1
1020 FLAG6=2
1030 PRINT "-------- 2. KLAN
G ----------"
1040 GOSUB 1920
1050 FLAG6=3
1060 PRINT "-------- 3. KLAN
G ----------"
1070 GOSUB 1920
1080 CALL SOUND(L(A),F1(A),0
,F2(A),0,F3(A),0)
1090 IF FLAG3=1 THEN 170
1100 GOTO 710
1110 IF FLAG12=1 THEN 170
1120 A=A-1
1130 M=A
1140 GOTO 170
1150 CALL CLEAR
1160 PRINT "KORREKTUR:.....
.....^^^^^^^^^^": :
1170 PRINT "WELCHE KORREKTUR
?.....->(1) LOESCHEN E
INER NOTE...->(2) ERGAENZEND
E NOTE.....->(3) VERAENDERN
 EINER NOTE": : :
1180 CALL KEY(0,K,S)
1190 IF S=0 THEN 1180
1200 IF K<49 THEN 1180
1210 IF K>51 THEN 1180
1220 ON K-48 GOTO 1350,1230,
1460
1230 M=M+1
1240 INPUT "BITTE GEBEN SIE
DIE NOTE AN,DIE NACH DER EIN
ZUFUEGENDEN NOTE STEHEN WIRD
. ":N
1250 FOR Y=M TO N+1 STEP -1
1260 Q=Y-1
1270 F1(Y)=F1(Q)
1280 F2(Y)=F2(Q)
1290 F3(Y)=F3(Q)
1300 L(Y)=L(Q)
1310 NEXT Y
1320 A=N
1330 FLAG3=1
```

```
1340 GOTO 1510
1350 INPUT "BITTE GEBN SIE D
IE NUMMER...DER NOTE AN, DIE
 GELOESCHT..WERDEN SOLL. ":N
1360 FOR A=N TO M
1370 I=A+1
1380 F1(A)=F1(I)
1390 F2(A)=F2(I)
1400 F3(A)=F3(I)
1410 L(A)=L(I)
1420 NEXT A
1430 M=M-1
1440 GOTO 170
1450 GOTO 1510
1460 INPUT "BITTE GEBEN SIE
NUMMER DER..NOTE EIN, DIE SI
E KORRI-....GIEREN WOLLEN. "
:A
1470 PRINT
1480 IF A<1 THEN 1460
1490 IF A>M THEN 1460
1500 IF INT(A)<>A THEN 1460
1510 PRINT :".......!.......
.....................v": :
1520 FLAG3=1
1530 GOTO 720
1540 CALL CLEAR
1550 PRINT "ABSPIELEN:......
............^^^^^^^^^^": : :
 : :
1560 FOR A=1 TO M
1570 CALL SOUND(L(A),F1(A),0
,F2(A),0,F3(A),0)
1580 PRINT "NOTENNUMMER: ";A
1590 NEXT A
1600 GOTO 170
1610 CALL CLEAR
1620 OPEN #1:"CS1",SEQUENTIA
L,INTERNAL,INPUT ,FIXED
1630 INPUT #1:M
1640 FOR A=1 TO M
1650 INPUT #1:L(A),F1(A),F2(
A),F3(A)
1660 NEXT A
1670 CLOSE #1
1680 GOTO 170
1690 CALL CLEAR
1700 OPEN #1:"CS1",SEQUENTIA
L,INTERNAL,OUTPUT,FIXED
1710 PRINT #1:M
1720 FOR A=1 TO M
1730 PRINT #1:L(A),F1(A),F2(
A),F3(A)
1740 NEXT A
1750 CLOSE #1
1760 GOTO 170
1770 CALL CLEAR
1780 PRINT "BITTE BEACHTEN S
IE, DASS BEIABSCHLUSS DES PR
OGRAMMS ALLEDATEN GELOESCHT
WERDEN!": : :
1790 PRINT "BITTE WAEHLEN SI
E:.........-->(1) ZURUECK Z
UR WAHLLISTE-->(2) ENDE DES
PROGRAMMS": : : : :
1800 CALL KEY(0,K,S)
1810 IF S=0 THEN 1800
1820 IF K<49 THEN 1800
1830 IF K>50 THEN 1800
1840 IF K=49 THEN 170
1850 CALL CLEAR
1860 PRINT "IN DIESEM FALL D
ARF ICH MICHBEI IHNEN FUER I
HR INTERESSEBEDANKEN UND MIC
H VERAB-....SCHIEDEN."
1870 PRINT "  ALEXANDER SABE
RSCHINKY '85": : : : : :
1880 CALL KEY(0,K,S)
1890 IF S=0 THEN 1880
1900 CALL CLEAR
1910 END
1920 PRINT "NOTENNR.:";A;"":
 :"KLANG(FREQUENZ)? "
1930 CALL KEY(0,K,S)
1940 IF S=0 THEN 1930
1950 IF K=131 THEN 2340
1960 IF K=132 THEN 2370
1970 IF K=133 THEN 2400
1980 IF K=134 THEN 2430
1990 IF K=135 THEN 2460
2000 IF K=129 THEN 2490
2010 IF K=136 THEN 2520
2020 IF K=67 THEN 2550
2030 IF K=68 THEN 2580
2040 IF K=69 THEN 2610
2050 IF K=70 THEN 2640
2060 IF K=71 THEN 2670
2070 IF K=65 THEN 2700
2080 IF K=72 THEN 2730
2090 IF K=96 THEN 2760
2100 IF K=9 THEN 2790
2110 IF K=11 THEN 2820
2120 IF K=123 THEN 2850
2130 IF K=125 THEN 2880
2140 IF K=124 THEN 2910
2150 IF K=191 THEN 2940
2160 IF K=159 THEN 2970
2170 IF K=181 THEN 3000
2180 IF K=182 THEN 3030
2190 IF K=183 THEN 3060
2200 IF K=158 THEN 3090
2210 IF K=57 THEN 3120
2220 IF K=53 THEN 3150
2230 IF K=54 THEN 3180
2240 IF K=55 THEN 3210
2250 IF K=56 THEN 3240
2260 IF K=15 THEN 3270
2270 IF K=14 THEN 3300
2280 IF K=12 THEN 3330
2290 IF K=1 THEN 3360
2300 IF K=6 THEN 3390
```

```
2310 IF K=80 THEN 3420
2320 IF K=83 THEN 32767
2330 GOTO 1920
2340 FRE=523
2350 C$="HOHES C"
2360 GOTO 3440
2370 FRE=587
2380 C$="HOHES D"
2390 GOTO 3440
2400 FRE=659
2410 C$="HOHES E"
2420 GOTO 3440
2430 FRE=698
2440 C$="HOHES F"
2450 GOTO 3440
2460 FRE=784
2470 C$="HOHES G"
2480 GOTO 3440
2490 FRE=880
2500 C$="HOHES A"
2510 GOTO 3440
2520 FRE=988
2530 C$="HOHES H"
2540 GOTO 3440
2550 FRE=262
2560 C$="MITTLERES C"
2570 GOTO 3440
2580 FRE=294
2590 C$="MITTLERES D"
2600 GOTO 3440
2610 FRE=330
2620 C$="MITTLERES E"
2630 GOTO 3440
2640 FRE=349
2650 C$="MITTLERES F"
2660 GOTO 3440
2670 FRE=392
2680 C$="MITTLERES G"
2690 GOTO 3440
2700 FRE=440
2710 C$="MITTLERES A"
2720 GOTO 3440
2730 FRE=494
2740 C$="MITTLERES H"
2750 GOTO 3440
2760 FRE=131
2770 C$="TIEFES C"
2780 GOTO 3440
2790 FRE=147
2800 C$="TIEFES D"
2810 GOTO 3440
2820 FRE=165
2830 C$="TIEFES E"
2840 GOTO 3440
2850 FRE=175
2860 C$="TIEFES F"
2870 GOTO 3440
2880 FRE=196
2890 C$="TIEFES G"
2900 GOTO 3440
2910 FRE=220
2920 C$="TIEFES A"
2930 GOTO 3440
2940 FRE=247
2950 C$="TIEFES H"
2960 GOTO 3440
2970 FRE=932
2980 C$="HOHES A#,B&"
2990 GOTO 3440
3000 FRE=554
3010 C$="HOHES C#,D&"
3020 GOTO 3440
3030 FRE=622
3040 C$="HOHES D#,E&"
3050 GOTO 3440
3060 FRE=740
3070 C$="HOHES F#,G&"
3080 GOTO 3440
3090 FRE=831
3100 C$="HOHES G#,A&"
3110 GOTO 3440
3120 FRE=466
3130 C$="MITTLERES A#,B&"
3140 GOTO 3440
3150 FRE=277
3160 C$="MITTLERES C#,D&"
3170 GOTO 3440
3180 FRE=311
3190 C$="MITTLERES D#,E&"
3200 GOTO 3440
3210 FRE=370
3220 C$="MITTLERES F#,G&"
3230 GOTO 3440
3240 FRE=415
3250 C$="MITTLERES G#,A&"
3260 GOTO 3440
3270 FRE=233
3280 C$="TIEFES A#,B&"
3290 GOTO 3440
3300 FRE=139
3310 C$="TIEFES C#,D&"
3320 GOTO 3440
3330 FRE=156
3340 C$="TIEFES D#,E&"
3350 GOTO 3440
3360 FRE=185
3370 C$="TIEFES F#,G&"
3380 GOTO 3440
3390 FRE=208
3400 C$="TIEFES G#,A&"
3410 GOTO 3440
3420 FRE=40000
3430 C$="PAUSE"
3440 PRINT TAB(7);C$;"": :
3450 ON FLAG4 GOTO 3460,3480
,3540
3460 F1(A)=FRE
3470 GOTO 3600
3480 IF FLAG5=1 THEN 3510
3490 F1(A)=FRE
3500 GOTO 3600
```

```
3510 F2(A)=FRE
3520 FLAG5=0
3530 GOTO 3600
3540 ON FLAG6 GOTO 3550,3570
,3590
3550 F1(A)=FRE
3560 GOTO 3600
3570 F2(A)=FRE
3580 GOTO 3600
3590 F3(A)=FRE
3600 IF FLAG2=1 THEN 3930
3610 PRINT "KLANGLAENGE? "
3620 CALL KEY(0,K,S)
3630 IF S=0 THEN 3620
3640 IF K=33 THEN 3720
3650 IF K=64 THEN 3750
3660 IF K=35 THEN 3780
3670 IF K=36 THEN 3810
3680 IF K=37 THEN 3840
3690 IF K=94 THEN 3870
3700 IF K=38 THEN 3900
3710 GOTO 3620
3720 L(A)=63
3730 C$="32-TEL"
3740 GOTO 3920
3750 L(A)=125
3760 C$="16-TEL"
3770 GOTO 3920
3780 L(A)=250
3790 C$=" 8-TEL"
3800 GOTO 3920
3810 L(A)=500
3820 C$=" 4-TEL"
3830 GOTO 3920
3840 L(A)=750
3850 C$=" 4-TEL PUNKTIERT"
3860 GOTO 3920
3870 L(A)=1000
3880 C$=" HALBE"
3890 GOTO 3920
3900 L(A)=2000
3910 C$=" GANZE"
3920 PRINT TAB(7);C$;: : :
3930 FLAG2=0
3940 RETURN
```

UNSER TELEFONSERVICE

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite
ACHTUNG !!!
Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13

BÖRSE

Drucke ihre Listings für 6,– DM pro Prg. Entferne Listschutz für 3,– DM pro Cassette !! Info 50 Pfg. anfordern. Matthias Orf, Birkenallee 34, D-3507 Baunatal 1

TI99/4A + Ext. Basic + orig. Cas.Rec. Kabel + TI-Bücher + umfangr. Software für 500,– DM zu verk. Tel. 040/6103696

Werden Sie Mitglied im großen TI-CLUB BAUNATAL! Tolles Clubheft, Prg. Speicher, Programmierlenkurs u.v.m. Info 50 Pfg. oder Clubheft 3,– DM anfordern! TI-CLUB BAUNATAL (TCB), Matthias Orf, Birkenaalee 34, D-3507 Baunatal, Tel. 0561/497990 !!!

Verkaufe originalverpackt: TI99/4A (19 Monate): 235,– DM TI99/4A + Rec-Kabel: 250,– DM Modul: Parsec: 45,– DM, Modul: Munch-Man: 45,– TI-Revue 1-7 (1a Zust.): 25,– TI-Revue (1–7) einzeln: 4,–. Andreas Braun, Tel. 02473/7320

TI99/4A zu verkaufen!! TI99/4A (wenig gelaufen) 235,– DM (+ Rec-Kabel: 250,–). Spiele-Modul: Parsec 45,– DM – Anruf ab 17 Uhr bei Herman-Josef Wollgarten, Tel. 02473/8628

TI99/4A + XB + Recorder + Kabel + Datenv. + Schach + Spielmodule + Joyst. + Literatur + Software DM 550,– 08461/1404

TI99 / 4A + ExBas + Peribox + RS232 + 32K + Disk + Edi/Ass + TI-Writer + Div. Module + Literatur 2500 DM. Apfeldorf 04122/54697 ab 18 Uhr

---

TI99/4A Verkaufe super Programme in TI-Ex-Basic. Info an Frank Henneberger, Pf. Legemann 19a, Nörvenich 2

TI-Writer, DM 210, Pers. Record. Keeping DM 40 Pers. Report Generator DM 40,– Statistik DM 40,–. 64K Platine u. Anleitung n. TI-Reven DM 25,– Tel. 0511/35 63 132 n. 18 h

---

Wer hat Lust, mit Anfänger Programme zu tauschen (TI-B)? Tel. 05971/57124). Holger Hammerschlag, 4440 Rheine 1

TI99+2Te Konsole + X-Bas+Rec. Kabel + Joy + div. Bücher 3 div. Progr. Cass. Preis VB 600 DM Schwetje 040/715 70 06

Suche Ex-Basic, Mini-Memory, ext. 32KB. Heinz Jörgensen, Ewensweg 2, 2987 Grossheide 1

TI-99/4A + Cass.-Rek. Kabel + Cass.-Rek. + Handbuch + Software Pries: VB 250,–. Tel: 02354/6974, 5882 Meinerzhagen 1

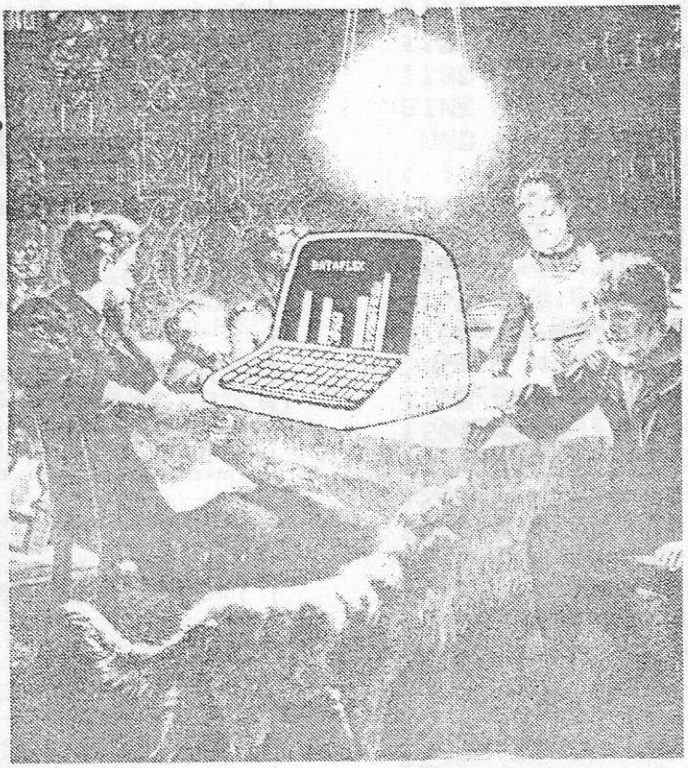
TI 99/4A: Verkaufe Buchungsjournal DM 70,– Bastert, Heinz Werner, 469 Herne 20, Tulpenweg 61

Suche Ext. Basic, evtl. mit Konsole, aber spottbillig und funktionstüchtig! Tel. 02103/44131

Suche komplette Peripherie, T. 09332/31 95

Tausche: HiFi-Cassettendeck Nakamichi-„Dragon" Reference bei AUDIO und STEREO PLAY Testberichte 8/83 Audio + 3/83 Stereoplay. Gegen: kompl. ausgestattete TI-99/4A Computer-Anlage. Mindest-Wert sollte nicht DM 3.500,– unterschreiten. Angebote an: R. Göhring, Alvesloher Str. 21, 2358 Kaltenkirchen, Tel. 04191/73 05

SPIEL SALON



See-Schlacht

Wir haben das Spiel „See-Schlacht" genannt, obwohl es sich eigentlich um ein U-Boot Spiel handelt! (es existieren bereits mehrere Spiele mit dem Namen „U-Boot") Das Spiel ist für zwei Personen gedacht, die ihre beiden Schiffe mit den Joysticks 1 und 2 steuern (geübten Programmierern wird es zwar nicht schwerfallen, das Spiel auf die Konsolentastatur umzuschreiben, was aber den Nachteil hat, daß zwei Spieler unabhängig voneinander die Tastatur bedienen müssen).

Die Aufgabe jedes Spielers besteht darin, den Gegner abzuschießen. Das Unterseeboot ist in jede Richtung steuerbar, genauso wie es in jede Richtung schießen kann. Geschossen wird, indem man auf den Feuerknopf drückt und innerhalb von ca. drei Sekunden den Joystick in die gewünschte Schußrichtung drückt. Besonders reizvoll ist es auch, wenn man Gänge in die Felswand schießt, um sich darin vor den Schüssen des Gegners zu schützen. Die Schüsse haben nur eine gewisse Reichweite und es kommt deshalb auf eine gute Reaktion im Nahkampf an. Das Anzeigenfeld rechts oben zeigt die restlichen Schüsse und die Siege des jeweiligen Spielers an. Gute Reaktion wird auch am Anfang verlangt, da es zu Spielbeginn nicht bekannt ist, wer welches Schiff zu steuern hat.

Zum Programm:

Um die einzelnen Steuerroutinen nicht doppelt schreiben zu müssen, haben wir eine zweidimensionale Variable verwendet, die angibt, welcher Spieler am Zuge ist. Dies bedeutet aber nicht, daß ein Joystick bevorzugt bzw. benachteiligt ist, da diese Variable laufend geändert wird.

Das Programm wurde möglichst einfach gehalten, um der Schnelligkeit des doch relativ langsamen TI Basic nichts zu nehmen.

Das Programm ist zusätzlich noch mit einigen REMarks erklärt. Das Spiel kann auch mit dem Xbasicmodul gefahren werden und wird dadurch wesentlich schneller.

Rühner + Rühmer

Apfelfänger

Haben Sie schon einmal in einer Fabrik als Apfelfänger gearbeitet? Nein? Dann versuchen Sie es doch mal mit Ihrem TI in der Grundversion und diesem Programm.

Sie müssen versuchen, die Äpfel, die übers Fließband laufen und irgendwo vom Band fallen, unten mit dem Korb aufzufangen. Ist Ihnen dieses gelungen, laufen Sie schnell zu einer der beiden Tonnen; rechts und links; und lassen Sie sich für den gefangenen Apfel Ihren Lohn gutschreiben. Als ordentlicher Fabrikarbeiter werden Sie natürlich nach dem jeweils gültigen Tarif bezahlt. Wenn Sie 10 Äpfel in der vorgegebenen Zeit zur Tonne gebracht haben, dürfen Sie am nächsten Tag weiterarbeiten. Ihr Tarif erhöht sich von Tag zu Tag um 50 %; die Zeit aber verkürzt sich jeden Tag um etwa 10 %.

Sollten Sie an einem Tag mit der Zeit nicht hinkommen, werden Sie mit Ihrem Lohn entlassen.

Steuern können Sie den Korb mit folgenden Tasten:
‚A' links
‚D' rechts
‚B' Funktionstaste

Die Taste ‚B' hat zwei Funktionen:

1. Wenn ein Apfel in der Luft ist, kann damit der Fall beschleunigt werden, wodurch man Zeit sparen kann. Der Nachteil ist aber, daß man den Korb nicht mehr bewegen kann, bis der Apfel unten ist. Daher sollte man schon sehr sicher sein, daß man unter dem Apfel steht, wenn man die Taste ‚B' benutzt.
2. Wenn kein Apfel in der Luft ist, kann man mit Taste ‚B' den nächsten Apfel kommen lassen.

Sollten Sie mehr als 4 Tage in der Fabrik aushalten, können Sie schon stolz auf Ihre Leistung sein.

Viel Spaß mit diesem Spiel wünscht

Torsten Rauer

Variable:

Die wichtigsten Variablen:
Q = die feste Zeit
F = Anzahl der noch zu fangenden Äpfel
G = Tarif
E = Guthaben
D = Noch vorhandene Zeit
B = 3 mögliche Positionen des fallenden Apfels

```
10 REM*********************
11 REM*                   *
12 REM*  APFELFAENGER     *
13 REM*  Copyright  by    *
14 REM*  Thorsten Rauer   *
15 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                   *
23 REM* Speicherbelegung *
24 REM*    4724 Bytes     *
25 REM*********************
26 REM
100 REM ***VORBEREITUNGEN
110 DIM B(3)
120 CALL CLEAR
130 CALL SCREEN(8)
140 FOR I=1 TO 12
150 CALL COLOR(I,2,8)
160 NEXT I
170 CALL COLOR(1,3,8)
180 CALL COLOR(9,5,8)
190 CALL COLOR(11,7,8)
200 CALL COLOR(12,14,8)
210 CALL CHAR(98,"00386CC6D6
C66C38")
220 CALL CHAR(106,"000000001
818FFFF")
230 CALL CHAR(115,"1A0C081C3
66E7E3C")
240 CALL CHAR(122,"387C44545
454447C")
250 CALL CHAR(107,"384444FEF
E7C7C38")
260 CALL CHAR(35,"FFFFFFFFFF
FFFFFF")
270 A=14
280 Q=700
290 D=Q
300 F=10
310 G=25
320 REM ***TITELBILD
330 PRINT : : : : : :" s s
 s s s s s s s s s s s s": : :
" s       APFELFAENGER      s"
: : :" s s s s s s s s s s s
 s s": : :
340 FOR I=1 TO 3
350 CALL SOUND(200,550,14,57
5,12)
360 CALL SOUND(200,550,12,57
5,10)
370 CALL SOUND(200,550,10,57
5,8)
380 CALL SOUND(800,210,4,221
,2)
390 NEXT I
400 PRINT ".s...c.THORSTEN.R
AUER....s": : :
410 PRINT " s s s s s s s s
 s s s s s":"jjjjjjjjjjjjjjj
jjjjjjjjjjj":"b....b....b..
..b....b....b": : :
420 PRINT " BITTE EINE TASTE
 DRUECKEN"
430 CALL KEY(0,K,S)
440 CALL SCREEN(14)
450 CALL SCREEN(3)
460 CALL SOUND(50,876,17,863
,19)
470 IF S=0 THEN 430
480 REM  ***SPIELAUFBAU
490 CALL CLEAR
500 FOR I=1 TO 13
510 READ AA,AB,AC,AD
520 CALL HCHAR(AA,AB,AC,AD)
530 NEXT I
540 FOR I=1 TO 2
550 READ A$,L,N
560 GOSUB 640
570 NEXT I
580 GOSUB 690
590 GOTO 840
600 DATA 1,1,35,96,5,1,106,3
2,6,1,98,1,6,5,98,1,6,9,98,1
,6,13,98,1,6,17,98,1,6,21,98
,1
610 DATA 6,25,98,1,6,29,98,1
,21,3,122,1,21,30,122,1,22,1
,35,96
620 DATA ZEIT######S,2,4,GUT
HABEN######DM###TARIF####DM,
23,2
630 REM ***PRINTROUTINE
640 FOR SS=1 TO LEN(A$)
650 CALL HCHAR(L,N-1+SS,ASC(
SEG$(A$,SS,1)))
660 NEXT SS
670 RETURN
680 REM ***ZAEHLER
690 CALL HCHAR(2,9,35,5)
700 FOR SS=1 TO LEN(STR$(D))
710 CALL HCHAR(2,8+SS,ASC(SE
G$(STR$(D),SS,1)))
720 NEXT SS
730 CALL HCHAR(2,17,35,16)
740 CALL HCHAR(2,20,115,F)
750 CALL HCHAR(23,11,35,5)
760 FOR SS=1 TO LEN(STR$(E))
770 CALL HCHAR(23,10+SS,ASC(
SEG$(STR$(E),SS,1)))
780 NEXT SS
790 CALL HCHAR(23,27,35,3)
800 FOR SS=1 TO LEN(STR$(G))
810 CALL HCHAR(23,26+SS,ASC(
SEG$(STR$(G),SS,1)))
820 NEXT SS
830 RETURN
840 REM ***BEGINN
850 CALL HCHAR(21,A,107)
```

```
 860 O=0
 870 RANDOMIZE
 880 B(2)=INT(RND*18+5)
 890 B(1)=B(2)-1
 900 B(3)=B(2)+1
 910 FOR I=1 TO B(2)
 920 CALL HCHAR(5,I,115)
 930 CALL SOUND(3,110,0,112,0
,1110,3)
 940 CALL HCHAR(5,I,106)
 950 NEXT I
 960 FOR I=0 TO 30 STEP 5
 970 CALL SOUND(-300,750,I)
 980 NEXT I
 990 REM***BEWEGUNGSABLAUEFE
1000 FOR I=7 TO 20
1010 IF O=1 THEN 1030
1020 R=INT(RND*3+1)
1030 CALL HCHAR(I,B(R),115)
1040 CALL SOUND(20,140,3,-1,
0)
1050 IF O<>0 THEN 1160
1060 CALL KEY(1,K,S)
1070 D=D-1
1080 IF (K=1)+(K=3)+(K=16)TH
EN 1090 ELSE 1160
1090 IF K<>16 THEN 1120
1100 O=1
1110 GOTO 1160
1120 CALL HCHAR(21,A,32)
1130 A=A+(K-2)
1140 IF (A<4)+(A>29)THEN 140
0
1150 CALL HCHAR(21,A,107)
1160 CALL HCHAR(I,B(R),32)
1170 NEXT I
1180 GOSUB 680
1190 IF A=B(R)THEN 1350
1200 FOR I=550 TO 110 STEP -
20
1210 CALL SOUND(-300,I,2)
1220 NEXT I
1230 CALL KEY(1,K,S)
1240 D=D-1
1250 IF D<=0 THEN 1760
1260 IF D>40 THEN 1280
1270 CALL SOUND(35,342,0)
1280 IF (K=1)+(K=3)+(K=16)TH
EN 1290 ELSE 1230
1290 IF K=16 THEN 850
1300 CALL HCHAR(21,A,32)
1310 A=A+(K-2)
1320 IF (A<4)+(A>29)THEN 140
0
1330 CALL HCHAR(21,A,107)
1340 GOTO 1230
1350 FOR AA=0 TO 30 STEP 10
1360 CALL SOUND(-400,1450,AA
)
1370 NEXT AA
1380 P=1
1390 GOTO 1230
1400 A=A-(K-2)
1410 IF P=1 THEN 1440
1420 IF I<20 THEN 1150 ELSE
1330
1430 REM ***TONNE MIT APFEL
       ***ERREICHT
1440 FOR I=585 TO 1150 STEP
25
1450 CALL SOUND(25,I,0)
1460 NEXT I
1470 E=E+G
1480 F=F-1
1490 P=0
1500 GOSUB 690
1510 IF F=0 THEN 1520 ELSE 1
330
1520 CALL SOUND(600,667,0,68
7,0)
1530 A$="PRIMA!DU HAST WIEDE
R"
1540 L=11
1550 N=6
1560 GOSUB 640
1570 A$="EINEN TAG GESCHAFFT
 !"
1580 L=13
1590 N=6
1600 P=0
1610 O=0
1620 GOSUB 640
1630 G=G+(G/100*50)
1640 IF G<>INT(G)THEN 1650 E
LSE 1670
1650 G=INT(G+1)
1660 E=E+(INT(D))
1670 Q=Q-(Q/100*10)
1680 D=Q
1690 F=10
1700 GOSUB 690
1710 CALL KEY(0,K,S)
1720 IF S=0 THEN 1710
1730 CALL HCHAR(11,1,32,32)
1740 CALL HCHAR(13,1,32,32)
1750 GOTO 850
1760 REM ***ENDE
1770 FOR I=1 TO 30 STEP 3
1780 CALL SOUND(150,110,I,35
0,I)
1790 NEXT I
1800 CALL CLEAR
1810 PRINT : : : : : : : : :
 :" LEIDER HABEN SIE AN DIES
EM": :" TAG NICHT SCHNELL GE
NUG": :" GEARBEITET.": :
1820 PRINT " DAHER MUESSEN W
IR SIE ": :" MIT IHREM BISHE
R ": :" ERARBEITETEN KAPITAL
 VON": :" ";E;"DM ENTLASSEN.
```

```
": :
1830 CALL KEY(0,K,S)
1840 CALL SCREEN(16)
1850 CALL SCREEN(9)
1860 CALL SOUND(25,110,0,110
0,0)
1870 IF S=0 THEN 1830
1880 CALL CLEAR
1890 PRINT " MOECHTEN SIE SI
CH ": :" NOCHEINMAL BEI UNS
BEWERBEN": : :" WENN JA DANN
 'J' DRUECKEN": : : :
1900 CALL KEY(0,K,S)
1910 CALL SCREEN(4)
1920 CALL SCREEN(5)
1930 CALL SOUND(25,110,5,-1,
0)
1940 IF S=0 THEN 1900
1950 IF K=74 THEN 1980
1960 CALL CLEAR
1970 END
1980 P=0
1990 O=0
2000 RESTORE 600
2010 GOTO 100
```

BÖRSE

GPL-Disassembler für XB+32K+Disk mit ausf. Anl. 20 DM. Deutsche Erklärung aller Forth-Wörter 20 DM. Suche Anleitung zum SST-Basic-Compiler. R. Prinzensing, Geitlingstr. 27, 4630 Bochum 6

Maschinenprogramme für XB, MM u. E/A: GPL-Disassembler, Fast-Copy etc. Info gg. Rückumschlag: Eichhorn, Mathias, Ziegelheck 1, 6240 Königstein

---

Tausche oder verkaufe Prog. in X-Basic oder Assembler. Überwiegend Anwenderprog. / Utilitys. Manfred Rüß, Tel. 030/363 26 55

LOGO-Interpreter ermögicht tolle Grafik Spielereien (XB) 20 DM an M. Bannert — Mozartstr. 5 8676 Schwrzenbach/Saale. Tel: 09284—8338

Verk. TI-Kochbuch 1 = 2 = DM 40,— TI-Tips & Tricks = DM 40. Joystickadapter = DM 25. M. Bannert. Tel. 09284-8338

Verk. TI99/4A + Ex-Basic + Parsec + Pec. Kabel + Flougsimulator + Prog. + Listings nur 495 DM. Neupr. 800 DM. Tel. 02671/14583

Ti99/4A + Ex-Basic + Rec-Kabel + Haushalts- und Schachmodul + 2 Bücher, VB 650 DM, nur kompl. Tel. 0211/55 45 09, nach 16 Uhr

---

TI-Module, Bücher, Drucker, Programme. 3S-Liste geg. Freiumschalg. Ralph Schad, Wasserbett 6, 68 Mannheim 51

Suche für den Ti: Spielmodule, Laufwerk, 32k Erweiterung, Schnittstelle extern oder wenn günstig in (mit) Peri-box. 0571/71597

```
10 REM********************
11 REM*                  *
12 REM*   SEE-SCHLACHT   *
13 REM*   Copyright by   *
14 REM* Ruhmer & Ruhmer  *
15 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*   Joystick 1&2   *
20 REM*                  *
21 REM*                  *
23 REM* Speicherbelegung *
24 REM*    5548 Bytes    *
25 REM********************
26 REM
220 CALL COLOR(12,16,10)
230 CALL CHAR(122,"F84444487
6514AC4")
240 CALL CLEAR
250 PRINT " ~~~.............
............~z~ - Ruhmer Sof
t...........~~~": :".......p
resents": : : : :
260 PRINT "*****************
************......See-Schlac
ht........*****************
***********": : : :
270 PRINT "COPYRIGHT BY Ruhm
er Manfred ": :"...........&
 Ruhmer Klaus"
280 GOSUB 1870
290 CALL KEY(1,K,T1)
300 CALL KEY(2,K,T)
310 IF (T=-1)+(T1=-1)THEN 31
5 ELSE 290
315 GOSUB 1950
320 CALL CLEAR
330 CALL CHAR(97,"FFFFFFFFFF
FFFFFF")
340 CALL CHAR(98,"80C0E0E0F0
FCFEFF")
350 CALL CHAR(99,"80C0C0C0C0
E0FEFF")
360 CALL CHAR(100,"FF7F3F3F1
F0F0701")
370 CALL CHAR(101,"FFFEFEFCF
CFCF080")
380 CALL CHAR(102,"01073F3F7
F7F7FFF")
390 CALL CHAR(120,"")
400 CALL CHAR(111,"3C3C3C7E9
9997E3C")
410 CALL CHAR(110,"000000181
8")
420 CALL CHAR(112,"0000001F1
0101010")
430 CALL CHAR(113,"000000F01
0101010")
440 CALL CHAR(114,"101010F0"
)
```

```
450 CALL CHAR(115,"1Ø1Ø1Ø1F"
)
460 CALL CHAR(116,"1Ø1Ø1Ø1Ø1
Ø1Ø1Ø1Ø")
470 CALL CHAR(117,"ØØØØØØFF"
)
480 CALL CHAR(118,"1Ø1Ø1Ø1F1
Ø1Ø1Ø1Ø")
490 CALL CHAR(119,"1Ø1Ø1ØFØ1
Ø1Ø1Ø1Ø")
500 SG(1)=Ø
510 SG(2)=Ø
520 CALL COLOR(10,16,6)
530 CALL COLOR(12,1,1)
540 CALL COLOR(8,3,3)
550 CALL COLOR(9,8,6)
560 CALL COLOR(1,6,6)
570 FOR N=1 TO 2
580 SU(N)=15
590 SH(N)=3
600 NEXT N
610 Z(1)=2
620 Z(2)=5
630 Z(3)=3
640 Z(4)=6
650 PRINT "^^^^^^^^^^^^^^^^^
^^^^puuuuuuuaaaaaaaaaaaaaaaa
aaa^tPL.1:1e...daae.........
.da^tSIEGE:"
660 PRINT ".....daac....fc..
..a^vuuuuuub.....daaaaaaaab.
..a^tPL.2:1ab.....daaaaaaaac
..a^tSIEGE:"
670 PRINT "aac.....de....daa
b a^suuuuuuaaa.............da
e a^^^^^^^^aaab.............
..daaaaaaaa"
680 PRINT "aaae.............
.....daaae aae..............
......daac ae.....fab....fab
.......aaab"
690 PRINT "......faaaab faaa
.......daaa.....faaaaaaaaaae
.........daa.....dae..daaaae.
...fc......"
700 PRINT ".................
..faab......................
..aaaaaaaaa.................
..daaaaaaaa"
710 PRINT "........fb.......
...daaaaaaa.......faab..fac.
......daae b.....faaaacfaae.
........da.."
720 PRINT "ac...faaaaaaaae..
..........faaaaaaaaaaaaaabfa
aaaaaaaaaaa"
730 CALL HCHAR(24,2,94,30)
740 CALL VCHAR(1,1,94,24)
750 CALL VCHAR(1,2,94)
760 CALL VCHAR(2,2,97,22)
770 CALL HCHAR(1,31,117)
780 CALL HCHAR(4,31,117)
790 CALL HCHAR(7,31,117)
800 CALL HCHAR(8,31,94)
810 CALL VCHAR(1,32,116,7)
820 CALL VCHAR(8,32,94,17)
830 CALL VCHAR(9,31,97,15)
840 CALL HCHAR(2,31,53)
850 CALL HCHAR(5,31,53)
860 CALL HCHAR(1,32,113)
870 CALL HCHAR(4,32,119)
880 CALL HCHAR(7,32,114)
890 RANDOMIZE
900 M=Ø
910 N=INT(RND*2)+1
920 M=M+1
930 A(N)=INT(RND*28)+3
940 B(N)=INT(RND*20)+3
950 CALL GCHAR(B(N),A(N),C)
960 IF C<>32 THEN 930
970 CALL HCHAR(B(N),A(N),111
)
980 A$=STR$(SG(N))
990 SP=31
1000 ZE=Z(N+2)
1010 GOSUB 1660
1020 GOSUB 1710
1030 IF M=1 THEN 920
1040 REM  * HAUPTPROGRAMM *
1050 CALL JOYST(N,X(N),Y(N))
1060 CALL KEY(N,K,T)
1070 IF K<>18 THEN 1100
1080 GOSUB 1200
1090 GOTO 1170
1100 IF (ABS(X(N))=Ø)*(ABS(Y
(N))=Ø)THEN 1170
1110 CALL GCHAR(B(N)-(Y(N)/4
),A(N)+(X(N)/4),C)
1120 IF C=32 THEN 1130 ELSE
1170
1130 CALL HCHAR(B(N)-(Y(N)/4
),A(N)+(X(N)/4),111)
1140 CALL HCHAR(B(N),A(N),32
)
1150 B(N)=B(N)-(Y(N)/4)
1160 A(N)=A(N)+(X(N)/4)
1170 GOSUB 1710
1180 GOTO 1050
1190 REM SCHUSSUNTERPROGRAMM
1200 R(N)=A(N)
1210 S(N)=B(N)
1220 FOR I=1 TO 16
1230 CALL JOYST(N,X(N),Y(N))
1240 IF (ABS(X(N))=4)+(ABS(Y
(N))=4)THEN 1270
1250 NEXT I
1260 RETURN
1270 ZE=Z(N)
1280 SP=30
1290 SU(N)=SU(N)-1
1300 IF SU(N)<Ø THEN 1540
```

```
1310 CALL SOUND(100,-6,0)
1320 A$=STR$(SU(N))
1330 IF LEN(A$)=2 THEN 1350
1340 A$=STR$(SU(N))&CHR$(120
)
1350 GOSUB 1660
1360 FOR I=1 TO 5
1370 CALL GCHAR(S(N)-(Y(N)/4
),R(N)+(X(N)/4),C)
1380 IF C=94 THEN 1260
1390 CALL HCHAR(S(N)-(Y(N)/4
),R(N)+(X(N)/4),110)
1400 S(N)=S(N)-(Y(N)/4)
1410 R(N)=R(N)+(X(N)/4)
1420 CALL HCHAR(S(N),R(N),32
)
1430 IF C<>111 THEN 1490
1440 GOTO 1530
1450 CALL HCHAR(B(N),A(N),32
)
1460 GOSUB 1710
1470 CALL HCHAR(B(N),A(N),32
)
1480 GOTO 890
1490 IF C<>32 THEN 1260
1500 NEXT I
1510 GOTO 1260
1520 REM * AUSWERTUNG *
1530 GOSUB 1710
1540 SH(N)=SH(N)-1
1550 GOSUB 1870
1560 IF SH(N)>0 THEN 1630
1570 GOSUB 1710
1580 SG(N)=SG(N)+1
1590 GOSUB 1870
1600 GOSUB 1870
1610 IF SG(N)>2 THEN 1760
1620 GOTO 570
1630 GOSUB 1710
1640 GOTO 1450
1650 REM * SCHREIBROUTINE *
1660 FOR I=1 TO LEN(A$)
1670 CALL HCHAR(ZE,SP,ASC(SE
G$(A$,I,1)))
1680 SP=SP+1
1690 NEXT I
1700 RETURN
1710 IF N=1 THEN 1740
1720 N=1
1730 GOTO 1750
1740 N=2
1750 RETURN
1760 CALL CLEAR
1770 FOR I=1 TO 8
1780 CALL COLOR(I,16,8)
1790 NEXT I
1800 CALL SCREEN(8)
1810 PRINT "PL.";N;"HAT GEWO
NNEN !!!":"----------------
-----": : : : :
1820 PRINT "NOCH EIN SPIEL G
EGEN DEN ": :"CHAMPION ? (J/
N)"
1830 CALL KEY(0,K,T)
1840 IF (K=74)+(K=106)THEN 5
00
1850 IF T=0 THEN 1830
1860 END
1870 CALL SOUND(120,262,8)
1880 CALL SOUND(70,110,30)
1890 CALL SOUND(120,262,8,33
0,6)
1900 CALL SOUND(70,110,30)
1910 CALL SOUND(120,262,7,33
0,5,392,3)
1920 CALL SOUND(70,110,30)
1930 CALL SOUND(450,330,3,39
2,2,523,0)
1940 RETURN
1950 CALL CLEAR
1960 PRINT "...*** SPIELANLE
ITUNG ***": : :"SIE BEFINDEN
 SICH TIEF UNTER": :"DER WAS
SEROBERFLAECHE IN": :
1970 PRINT "EINEM U-BOOT DAS
 SIE MIT": :"DEM JOYSTICK ST
EUERN MUESSEN": :"ES SPIELEN
 2 SPIELER DIE": :
1980 PRINT "IHRE U-BOOTE JEW
EILS MIT": :"JOYSTICH 1 ODER
 JOYSTICK 2": :"STEUERN"
1990 PRINT : :"T A S T E  D
R U E C K E N "
2000 CALL KEY(1,T,U)
2010 CALL KEY(2,T1,U1)
2020 IF (U=0)*(U1=0)THEN 200
0
2030 CALL CLEAR
2040 PRINT "SIE MUESSEN NUN
VERSUCHEN": :"DEN GEGNER ABZ
USCHIESSEN": :"GESCHOSSEN WI
RD INDEM MAN": :
2050 PRINT "AUF DIE ACTION-T
ASTE DRUECKT": :"UND SOFORT
DEN JOYSTICK IN": :"DIE GEWU
ENSCHTE SCHUSS-": :
2060 PRINT "RICHTUNG DRUECKT
": : :".....*** VIEL SPASS *
**": :"T A S T E  D R U E C
K E N"
2070 CALL KEY(1,T,U)
2080 CALL KEY(2,T1,U1)
2090 IF (U=0)*(U1=0)THEN 207
0
2100 CALL CLEAR
2110 RETURN
```

Neue Telefon-Nummer!
089/1298013 Ab 1.10.85!

BUNTE MISCHUNG:

Pokelisten-generator

Das Programm dient zur Erstellung von Pokelisten auf einem Drucker. Es fragt zunächst die Schnittstellenbezeichnung sowie Anfangs- und Endadresse des gewünschten Bereiches ab. Anschließend wird die Pokeliste in DATA-Zeilen ausgedruckt. Ein im Speicher befindliches Assembler-Programm kann dadurch in eine abschreibfähige Form gebracht und später in einem BASIC-Programm verwendet werden.

Karnaugh-Veitch-Diagramm

Dieses Programm dient zur Ermittlung von Schaltfunktionen aus einer Wahrheitstabelle. Dabei wird zunächst eine Wahrheitstabelle für vier Variable erstellt und die Eingabe des jeweiligen Zustandes für A abgefragt. Bei einer fehlerhaften Eingabe kann durch Betätigen der Taste K der zuletzt eingegebene Wert korrigiert werden. Mit x werden die sogenannten don't care Felder gekennzeichnet. Sie werden dann verwendet, wenn der Zustand des Ausgangs A für die Berechnung der Schaltfunktion ohne Bedeutung ist. Nach Abschluß des Eingabevorganges erstellt das Programm das sogenannte KV-Diagramm, indem es die Werte für A bestimmten Feldern des Diagramms zuordnet. Das KV-Diagramm ist die grafische Veranschaulichung der Wahrheitstabelle einer Schaltfunktion. Dieses Verfahren der Vereinfachung von Schaltfunktionen wurde 1952 von E.W. Veitch erfunden und 1953 von M. Karnaugh weiterentwikkelt. Das Programm ermittelt nun aus dem Diagramm die Schaltfunktion, indem es mit 1 gekennzeichnete Felder zu Gruppen zusammenfaßt und diesen ein bestimmtes Ergebnis zuordnet.

Dieses Programm kann in zwei Arten angewendet werden:
Zum einen können vorhandene Schaltungen minimiert werden. Dazu wird aus der vorhandenen Schaltung die Wahrheitstabelle ermittelt und eingegeben. Für Eingangskombinationen, die nicht vorkommen oder nicht von Bedeutung sind, wird x eingegeben. Das Programm ermittelt aus der Eingabe die optimierte Schaltfunktion, aus der die Schaltung entworfen werden kann.

Zum anderen kann aus einer entwickelten Wahrheitstabelle die minimierte Schaltfunktion ermittelt werden. Die Eingabe erfolgt analog zum o.a. Beispiel.

Bernhard Schmidt

Hexdump

Mit diesem Programm kann der Speicherinhalt auf einem Drucker ausgegeben werden. Es werden jeweils 16 Werte in hexadezimaler Form und die Hex-Adresse des jeweils ersten Wertes ausgegeben.

Bernard Schmidt

Widerstand

Das Programm „Widerstand" ist für den Elektronik-Bastler gedacht.

Im Menue kann folgende Auswahl getroffen werden.
0 – Errechnen von parallelgeschalteten Widerständen zu einem Ersatzwiderstand 2, 3, 4 und 5 parallele können berechnet werden.
1 und 2 berechnet der Farbcode
z.B. 0 = schwarz oder rot = 2
3. 4 u. 5 errechnen wahlweise Widerstand, Strom oder Spannung.
6. ist das Programmende.

Rüdiger Hussy

Space Invaders

Der alte Spielhallenhit „Space Invaders" hat nun auch endlich den TI99 eingeholt. Zwar eine etwas abgeänderte Version, auch etwas leichter zu spielen, aber unverkennbar standen die Original-Invaders für dieses Programm Pate. Zu den Spielregeln gibt es da wohl nicht mehr viel zu sagen. Jeder Computerfreak ist mit diesem Oldie vertraut. Daß wir es trotzdem abdrucken, hat drei Gründe. Erstens hat sich der Autor viel Mühe damit gemacht und das soll natürlich auch belohnt werden, denn dies ist Eure Zeitschrift und wir weisen nur manchmal ein bißchen in die Richtung, wo's lang gehen sollte. Zweitens ist das Spiel in TI-Basic geschrieben worden. Damit denken wir an die User, die („immer noch?") kein Extended-Basic Modul besitzen (Jetzt wird's aber langsam Zeit). Und drittens ist die Grafik ganz toll geworden. Auch die Spielgeschwindigkeit ist für die mageren Möglichkeiten des TI-Basic hervorragend programmiert. Zum Spielen nehmt Ihr übrigens Euren Joystick aus der Schublade. Damit läßt sich die Kanone nach rechts und links bewegen.

Prognose-Programm

Sinn: Dieses Programm errechnet aufgrund von max. 22 feststehenden Werten bis zu 22 Werte in die Zukunft (Prognose oder Hochrechnung).

Diese Werte werden auf dem Bildschirm ausgegeben. Anschließend werden diese Hochrechnungswerte auf einer Grafik dargestellt.
100–160 Dimensionierung und Titel
170 Einleitung
180–300 Abfrage der feststehenden Werte
310–430 Ausgabe der Prognosewerte (Hochrechnung)
440–820 Ausgabe der Grafik
830–900 Ende des Programms
910–950 Unterprogramm Stotterdruck

Bitte lesen Sie weiter auf Seite 66

```
10 ! ************************
11 ! *                      *
12 ! *      PROGNOSE        *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *     R. Bitter        *
17 ! *                      *
18 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     4447 Bytes       *
28 ! *                      *
29 ! ************************
30 !
100 CALL CLEAR :: DIM W(22),
DAT(22),WE(22):: CALL CHAR(3
3,"0000000FFFF")
110 W$="PROGNOSE-PROGRAMM"
120 CALL STOTT(12,W$)
130 ZE1=11 :: SP1=8 :: ZE2=1
3 :: SP2=26
140 CALL RAHMEN(ZE1,SP1,ZE2,
SP2)
150 DISPLAY AT(24,3):"progra
mmiert von r.bitter"
160 FOR I=1 TO 400 :: NEXT I
170 CALL EINLEITUNG
180 CALL CLEAR
190 DISPLAY AT(1,1):"WIEVIEL
 WERTE LIEGEN VOR:" :: DISPL
AY AT(2,1):"MAX. 22 WERTE" ;
: ACCEPT AT(1,27):N :: CALL
HCHAR(2,3,33,28)
200 IF N>22 THEN 190
210 FOR I=1 TO N
220 DISPLAY AT(I+2,3):"WERT"
 :: DISPLAY AT(I+2,8):USING
"##":I :: DISPLAY AT(I+2,10)
:":" :: ACCEPT AT(I+2,15):W(
I)
230 NEXT I
240 T=LOG(W(1))
250 X=0
260 FOR I=2 TO N :: L=LOG(W(
I)):: T=T+L :: X=X+(I-1)*L :
: NEXT I
270 A=6*(2*X/(N-1)-T)/N/(N+1
)
280 G=EXP(A)-1 :: G=100*(EXP
(A)-1):: G=INT(G*100+.5)/100
290 DISPLAY AT(24,1):"MITTL.
WACHSTUMSRATE:" :: DISPLAY A
T(24,22):USING "###.##":G ::
 CALL HCHAR(24,31,37)
300 CALL KEY(0,KE,S):: IF S=
0 THEN 300
310 CALL CLEAR
320 DISPLAY AT(1,1):"HOCHREC
HNUNGS-WERTE?" :: DISPLAY AT
(2,1):"MAXIMAL 22 WERTE" ::
ACCEPT AT(1,25):M :: CALL HC
HAR(2,3,33,28)
330 G=EXP(A)-1 :: WT=EXP(T/N
-A*(N-1)/2)
340 CALL HCHAR(1,3,32,28)::
DISPLAY AT(1,2):"HOCHRECHNUN
G DER PERIODEN"
350 K=N :: H=(G+1)^K
360 FOR I=1 TO M
370 WE(I)=WT*H :: WE(I)=INT(
WE(I)*100+.5)/100
380 DISPLAY AT(I+2,3):"WERT"
 :: DISPLAY AT(I+2,8):USING
"##":I :: DISPLAY AT(I+2,10)
:":" :: DISPLAY AT(I+2,15):U
SING "######":WE(I)
390 K=K+1
400 H=(G+1)^K
410 IF I=M THEN 430
420 NEXT I
430 CALL KEY(0,KE,S):: IF S=
0 THEN 430
440 CALL CLEAR :: CALL SCREE
N(16)
450 DISPLAY AT(24,1):"HOCHRE
CHNUNG:";M;"PERIODEN"
460 S1$="7F7F7F7F7F7F7F7F"
470 S2$="007F7F7F7F7F7F7F00
07F7F7F7F7F7F00000007F7F7F7F
F000000007F7F7F7F"
480 S3$="00000000007F7F7F00
0007F00007F7F000000000000007
F000000000000000"
490 CALL CHAR(96,S1$):: CALL
 CHAR(97,S2$):: CALL CHAR(10
1,S3$)
500 S4$="F8080808080808080F0
0000000000000FF0000000000000
0380808080808080" :: CALL C
HAR(112,S4$):: CALL COLOR(11
,13,1)
510 S5$="F000000000000000001F1
010101010101C1010101010101
0" :: CALL CHAR(116,S5$)
520 DISPLAY AT(10,1):"IST 10
0% SUMME ODER MAXIMUM": :"DE
R DATEN ? (S/M)"
530 ACCEPT AT(14,14):MOD$
540 IF MOD$<>"S" AND MOD$<>"
M" THEN 530
550 FOR I=10 TO 14 :: CALL H
CHAR(I,3,32,28):: NEXT I
560 CALL COLOR(9,7,1,10,7,1)
570 FOR I=0 TO 10 :: DISPLAY
 AT(2*I+2,1):USING "###":(10
-I)*10 :: NEXT I
580 FOR I=1 TO 10 :: DISPLAY
```

```
 AT(2*I,4):"p" :: DISPLAY AT
(2*I+1,4):"s"
590 DISPLAY AT(2*I,27):"u" :
: DISPLAY AT(2*I+1,27):"v" :
: NEXT I
600 DISPLAY AT(22,4):"q"
610 FOR I=1 TO 11 :: DISPLAY
 AT(22,2*I+3):"rr" :: NEXT I
620 DISPLAY AT(22,27):"t"
630 DA=WE(1)
640 FOR I=2 TO M
650 DA=MAX(DA,WE(I))
660 NEXT I
670 IF MOD$="S" THEN 740
680 DISPLAY AT(1,27):"MA"
690 FOR I=1 TO M
700 WE(I)=WE(I)/DA*160
710 WE(I)=INT(WE(I)+.5)
720 NEXT I
730 GOTO 760
740 DA=0 :: DISPLAY AT(1,27)
:"SU" :: FOR I=1 TO M :: DA=
DA+WE(I):: NEXT I
750 GOTO 690
760 FOR I=1 TO M
770 HOEHE=INT(WE(I)/8)
780 ZUS=WE(I)-HOEHE*8
790 CALL VCHAR(21-HOEHE,I+6,
104-ZUS,1)
800 CALL VCHAR(22-HOEHE,I+6,
96,HOEHE)
810 NEXT I
820 CALL KEY(0,KE,S):: IF S=
0 THEN 820
830 W$="E N D E"
840 CALL CLEAR
850 CALL STOTT(12,W$)
860 ZE1=11 :: SP1=13 :: ZE2=
13 :: SP2=21
870 CALL RAHMEN(ZE1,SP1,ZE2,
SP2)
880 DISPLAY AT(24,3):"PROGRA
MMIERT VON R.BITTER"
890 FOR I=1 TO 800 :: NEXT I
900 END
910 SUB STOTT(12,W$)
920 Q=INT(32-LEN(W$))/2+1
930 FOR AX=Q TO LEN(W$)+Q-1
:: S$=SEG$(W$,AX-Q+1,1):: FO
R AZ=31 TO AX STEP -1
940 CALL HCHAR(12,AZ+1,32)::
 CALL HCHAR(12,AZ,ASC(S$))::
 NEXT AZ :: NEXT AX
950 SUBEND
960 SUB RAHMEN(ZE1,SP1,ZE2,S
P2)
970 CALL CHAR(128,"0000000FFF
F",129,"1818181818181818",13
0,"0000001F1F181818",131,"00
0000F8F8181818",132,"1818181
F1F",133,"181818F8F8")
980 CALL HCHAR(ZE1,SP1,130):
: CALL HCHAR(ZE1,SP2,131)::
CALL HCHAR(ZE2,SP1,132):: CA
LL HCHAR(ZE2,SP2,133)
990 CALL HCHAR(ZE1,SP1+1,128
,SP2-SP1-1)
1000 CALL HCHAR(ZE2,SP1+1,12
8,SP2-SP1-1)
1010 CALL VCHAR(ZE1+1,SP1,12
9,ZE2-ZE1-1):: CALL VCHAR(ZE
1+1,SP2,129,ZE2-ZE1-1)
1020 SUBEND
1030 SUB EINLEITUNG
1040 CALL CLEAR
1050 ZE1=1 :: SP1=2 :: ZE2=2
4 :: SP2=31
1060 CALL RAHMEN(ZE1,SP1,ZE2
,SP2)
1070 DISPLAY AT(2,8):"HOCHRE
CHNUNGEN" :: CALL HCHAR(3,10
,33,14)
1080 DISPLAY AT(4,1):"DIESES
 PROGRAMM BERECHNET": :"AUF
GRUND VON MAX.22 FEST-": :"S
TEHENDEN WERTEN BIS ZU 22":
:"WERTE IN DIE ZUKUNFT"
1090 DISPLAY AT(12,1):"(PROG
NOSE).VORHER WIRD DIE": :"MI
TTLERE WACHSTUMSRATE,AUF": :
"GRUND DER EINGEGEBENEN WERT
E"
1100 DISPLAY AT(18,1):"ANGEZ
EIGT.JE MEHR WERTE EIN-": :"
GEGEBEN WERDEN UM SO GENAUER
": :"DIE PROGNOSE.
   >>>"
1110 CALL KEY(0,KE,S):: IF S
=0 THEN 1110
1120 FOR I=4 TO 23 :: CALL H
CHAR(I,3,32,28):: NEXT I
1130 DISPLAY AT(4,1):"NACH D
ER AUSGABE DER PROG-": :"NOS
E KANN DURCH DRUECKEN": :"EI
NER TASTE DIE ERRECHNETEN":
:"WERTE AUF EINER BALKEN-"
1140 DISPLAY AT(12,1):"GRAFI
K AUSGEGEBEN WERDEN."
1150 DISPLAY AT(22,26):">>>"
1160 CALL KEY(0,KE,S):: IF S
=0 THEN 1160
1170 SUBEND
```

Fortsetzung von Seite 64

960–1020 Unterprogramm Rahmen
1030–1170 Unterprogramm Einleitung

Anm. der Redaktion: Der Programmierstil deutet darauf hin, daß der Autor noch nicht allzulange im Besitz des Xbasic-Moduls ist.

TI-Basic und Xbasic-Befehle sind bunt durcheinander gemischt. Durch eine korrekte Programmierung in Xbasic wäre

Bitte lesen Sie weiter auf Seite 69

```
10 ! ********************
11 ! *                  *
12 ! *   WIDERSTAENDE   *
13 ! *                  *
14 ! *   Copyright by   *
15 ! *  Ruediger Huesy  *
16 ! *                  *
17 ! *                  *
19 ! * Benoetigte Geraete *
20 ! *  T199/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     3891 Bytes     *
28 ! *                    *
29 ! ********************
30 !
290 ON BREAK NEXT :: ON WARN
ING NEXT
300 CALL CLEAR :: CALL SCREE
N(2)
310 X=X+1 :: CALL COLOR(X,11
,2):: IF X=14 THEN 330
320 GOTO 310
330 CALL CLEAR
340 PRINT "...W I D E R S T
A N D....."
350 PRINT :: PRINT
360 PRINT " @...BY...R U E D
 I G E R --.................
.............H U S S Y.."
370 PRINT :: PRINT :: PRINT
:: PRINT :: PRINT "BITTE TAS
TE DRUECKEN"
380 PRINT :: PRINT :: PRINT
:: PRINT :: PRINT
390 CALL KEY(0,K,S)
400 IF S=0 THEN 390
410 !
510 !
520 CALL CLEAR
530 PRINT "......M E N U E"
540 Y=0
550 PRINT :: Y=Y+1 :: IF Y=4
 THEN 570
560 GOTO 550
570 PRINT "0 FUER PARALELLSC
HALTUNG" :: PRINT :: PRINT "
1 FUER WIDERSTAND >>>>>.....
..CODFARBE" :: PRINT
580 PRINT "2 FUER CODFARBE..
>>>>>>>.....WIDERSTAND" ::
PRINT
590 PRINT "3 ERRECHNEN VON S
PANNUNG" :: PRINT
600 PRINT "4 ERRECHNEN VON S
TROM" :: PRINT
610 PRINT "5 ERRECHNEN VON W
IDERSTAND" :: PRINT
620 PRINT "6 PROGRAMMENDE" :
: PRINT
630 INPUT "IHRE WAHL? ":A$ :
: IF A$="1" THEN 1110 :: IF
A$="6" THEN 950 :: IF A$="0"
 THEN 2000
640 IF A$="2" THEN 960 :: IF
 A$="5" THEN 740 :: IF A$="4
" THEN 650 :: IF A$="3" THEN
 850 :: IF A$>"6" THEN 520
650 CALL CLEAR :: PRINT "ERR
ECHNEN VON STROM" :: PRINT :
: PRINT
660 INPUT "SPANNUNG IN VOLT(
V) ":U
670 INPUT "WIDERSTAND IN OHM
 ":R
680 I=U/R :: PRINT :: PRINT
:: PRINT "ES FLIESSEN":I:"AM
PERE(A)"
690 PRINT :: PRINT :: PRINT
:: INPUT "UMRECHNEN IN mA J/
N":K$
700 IF K$="J" THEN 710 :: IF
 K$="N" THEN 720
710 MA=I*1000 :: PRINT "DAS
SIND":MA:"MILLI AMPERE"
720 PRINT :: PRINT :: PRINT
"TASTE DRUECKEN"
730 CALL KEY(0,K,S):: IF S=0
 THEN 730 :: GOTO 520
740 CALL CLEAR :: PRINT :: P
RINT :: PRINT "ERRECHNEN  VO
N WIDERSTAND " :: PRINT :: P
RINT
750 INPUT "SPANNUNS (V) ":C
760 INPUT "STROM (A) ":ER
770 R=C/ER :: PRINT :: PRINT
 :: PRINT "ES SIND":R:"OHM"
780 PRINT :: PRINT :: PRINT
:: INPUT "UMRECHNEN OHM IN K
ILO OHM J/N":L$
790 IF L$="J" THEN 810 :: IF
 L$="N" THEN 820
800 !
810 KO=R/1000 :: PRINT "DAS
SIND":KO:"KILO OHM"
820 PRINT :: PRINT :: PRINT
"BITTE EINE TASTE DRUECKEN"
830 CALL KEY(0,K,S):: IF S=0
 THEN 830 :: GOTO 520
840 !
850 CALL CLEAR :: PRINT :: P
RINT :: PRINT "ERRECHNEN VON
 SPANNUNG" :: PRINT :: PRINT
860 INPUT "WIDERSTAND IN OHM
 ":WI :: INPUT "STROM IN AMP
ERE ":AM
870 REM
880 UI=WI*AM :: PRINT :: PRI
NT :: PRINT "ES IST EINE SPA
NNUNG VON":UI:"VOLT"
```

```
890 PRINT :: PRINT :: INPUT
"UMRECHNEN IN MV J/N ":VB$
900 IF VB$="J" THEN 920 :: I
F VB$="N" THEN 930
910 REM
920 MV=UI*1000 :: PRINT :: P
RINT :: PRINT "DAS SIND":MV:
"MILLI VOLT"
930 PRINT :: PRINT :: PRINT
"BITTE EINE TASTE DRUECKEN"
940 CALL KEY(0,K,S):: IF S=0
 THEN 940 :: GOTO 520
950 END
960 CALL CLEAR :: PRINT "COD
FARBE > WIDERSTANDSW." :: IN
PUT "CODFARBEN: ":AX$ :: PRI
NT :: PRINT :: PRINT "DIE DR
ITTE RINGFARBE SIND NULLEN"
970 IF AX$="SCHWARZ" THEN 10
80 :: IF AX$="BRAUN" THEN 10
70 :: IF AX$="ROT" THEN
1060 :: IF AX$="ORANGE" THEN
 1050 :: IF AX$="GELB" THEN
1040
980 IF AX$="GRUEN" THEN 1030
 :: IF AX$="BLAU" THEN 1020
:: IF AX$="VIOLETT" THEN 101
0 :: IF AX$="GRAU" THEN 1000
 :: IF AX$="WEISS" THEN 990
990 PRINT "9" :: GOTO 1090
1000 PRINT "8" :: GOTO 1090
1010 PRINT "7" :: GOTO 1090
1020 PRINT "6" :: GOTO 1090
1030 PRINT "5" :: GOTO 1090
1040 PRINT "4" :: GOTO 1090
1050 PRINT "3" :: GOTO 1090
1060 PRINT "2" :: GOTO 1090
1070 PRINT "1" :: GOTO 1090
1080 PRINT "0" :: GOTO 1090
1090 PRINT :: PRINT :: INPUT
 "NOCHMAL ? ":KL$
1100 IF KL$="J" THEN 960 ::
IF KL$<>"J" THEN 520
1110 CALL CLEAR :: PRINT "WI
DERSTANDSW.>CODFARBE" :: PRI
NT :: PRINT
1120 PRINT :: PRINT "DER DRI
TTE RINGWERT IST DIE ANZAHL
DER NULLEN" :: PRINT :: PRIN
T
1130 PRINT :: PRINT " EINGAB
E DES RINGWERTES" :: INPUT A
S$
1140 IF AS$="0" THEN 1250 ::
 IF AS$="1" THEN 1240 :: IF
AS$="2" THEN 1230 :: IF AS$=
"3" THEN 1220 :: IF AS$="4"
THEN 1210 :: IF AS$="5" THEN
 1200
1150 IF AS$="6" THEN 1190 ::
 IF AS$="7" THEN 1180 :: IF
AS$="8" THEN 1170 :: IF AS$=
"9" THEN 1160

1160 PRINT "WEISS"
1170 PRINT "GRAU" :: GOTO 12
40

1180 PRINT "VIOLETT" :: GOTO
 1260
1190 PRINT "BLAU" :: GOTO 12
60

1200 PRINT "GRUEN" :: GOTO 1
260
1210 PRINT "GELB" :: GOTO 12
60
1220 PRINT "ORANGE" :: GOTO
1260
1230 PRINT "ROT" :: GOTO 126
0
1240 PRINT "BRAUN" :: GOTO 1
260
1250 PRINT "SCHW." :: GOTO 1
260
1260 PRINT :: PRINT :: INPUT
 "NOCHEINMAL ? ":BN$
1270 IF BN$="J" THEN 1110 ::
 IF BN$<>"J" THEN 520
2000 CALL CLEAR :: PRINT "ER
RECHNEN DES ERSATZWIDER - ST
ANDES VON PARALELLSCHALT- UN
GEN" :: PRINT :: PRINT
2010 INPUT "WIEVIELE WIDERST
AENDE SIND..PARALELL? (2,3,4
,5) ":KL$ :: PRINT :: PRINT
2020 IF KL$="2" THEN 2050 ::
 IF KL$="3" THEN 2070 :: IF
KL$="4" THEN 2100 :: IF KL$=
"5" THEN 2200 :: IF KL$>"5"
THEN 2010 :: IF KL$<"2" THEN
 2010
2030 !
2040 !
2050 INPUT "1.WERT ":Q :: IN
PUT "2.WERT ":W
2060 Z=(Q*W)/(Q+W):: PRINT "
DER ERSATZWERT IST":Z :: PRI
NT "OHM" :: FOR K=1 TO 500 :
: NEXT K :: GOTO 520
2070 INPUT "WERT 1. ":V :: I
NPUT "WERT 2. ":B :: INPUT "
WERT 3. ":N :: M=(1/V)+(1/B)
+(1/N):: L=1/M
2080 PRINT "DER ERSATZWERT I
ST ":L :: PRINT "OHM" :: PRI
NT :: PRINT :: PRINT "BITTE
EINE TASTE DRUECKEN"
2090 CALL KEY(0,K,S):: IF S=
0 THEN 2090 :: GOTO 520
2100 INPUT "WERT 1. ":S :: I
NPUT "WERT 2. ":Z :: INPUT "
WERT 3. ":X :: INPUT "WERT 4
. ":C
```

```
2110 N=(1/S)+(1/Z)+(1/X)+(1/
C):: L=1/N
2120 PRINT "DER ERSATZWERT I
ST ";L :: PRINT "OHM" :: PRI
NT :: PRINT :: PRINT "BITTE
EINE TASTE DRUCKEN"
2130 CALL KEY(0,K,S)
2140 IF S=0 THEN 2130
2150 GOTO 520
2160 !
2190 !
2200 INPUT "WERT 1. ":G :: I
NPUT "WERT 2. ":H :: INPUT "
WERT 3. ":J :: INPUT "WERT 4
. ":K :: INPUT "WERT 5. ":L
2210 U=(1/G)+(1/H)+(1/J)+(1/
K)+(1/L):: A=1/U
2220 PRINT "DER ERSATZWERT I
ST ";A :: PRINT "OHM" :: PRI
NT :: PRINT :: PRINT "BITTE
EINE TASTE DRUCKEN"
2230 CALL KEY(0,K,S)
2240 IF S=0 THEN 2230
2250 GOTO 520
```

Fortsetzung von Seite 66

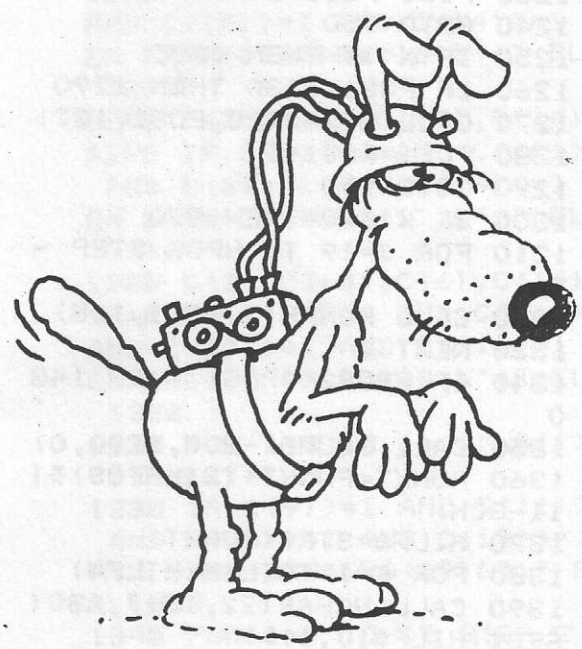
das Programm wesentlich kürzer ausgefallen.

Trotzdem haben wir uns aus zwei Gründen zu einem Abdruck entschlossen!

Zum ersten fällt es aufgrund der vielen TI-Basic-Befehle nicht mehr allzu schwer, das Programm komplett in TI-Basic umzuschreiben und zum anderen bietet das vorliegende Programm doch eine beachtliche Hilfestellung beim Errechnen der benötigten Werte. Die Formeln und Ergebnisse wurden von uns überprüft und können unbedenklich übernommen werden.

Alle Angaben sind korrekt.

H. König



```
10 REM*********************
11 REM*                   *
12 REM* SPACE  INVADERS  *
13 REM*  Copyright  by    *
14 REM* Sascha Dierkes    *
15 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                   *
23 REM* Speicherbelegung *
24 REM*    5139 Bytes     *
25 REM*********************
26 REM
180 RANDOMIZE
190 RESTORE
200 CALL CLEAR
210 PRINT "*SPACE INVADERS*"
: :"SCHIESSEN SIE SO VIELE..
....BOMBEN WIE MOEGLICH AB..
....DESTO EHER SIE DIESE"
220 PRINT "TREFFEN  DESTO  M
EHR.....PUNKTE GIBT ES.":
 :"BERUEHREN DIE BOMBEN JEDO
CH DEN BODEN HABEN SIE EINE.
..STATION VERLOREN"
230 PRINT "UNTEN WERDEN DIE
ANZAHL DER RESTLICHEN STATIO
NEN SOWIE..IHRE PUNKTE UND A
NGRIFFS-...WELLE DER INV.";
240 PRINT " ANGEZEIGT."
250 CALL KEY(0,K,S)
260 IF S=0 THEN 250
270 CALL CLEAR
280 PRINT "JOYSTICK VERSION"
: : :
290 INPUT "ANZAHL DER ANGREI
FER PRO W. (2-10)":ANZ
300 IF (ANZ<2)+(ANZ>10)+(ANZ
<>INT(ANZ))THEN 290
310 CALL CLEAR
320 FOR I=3 TO 4
330 CALL COLOR(I,4,2)
340 NEXT I
350 FOR I=10 TO 13
360 CALL COLOR(I,1,1)
370 NEXT I
380 CALL COLOR(2,1,1)
390 FOR I=1 TO 50
400 ZE=INT(RND*9)+1
410 SP=INT(RND*30)+1
420 CALL HCHAR(ZE,SP,46,1)
430 NEXT I
440 CALL CHAR(124,"0103070F1
F3F7FFF")
450 CALL CHAR(125,"80C0E0F0F
8FCFEFF")
460 CALL CHAR(111,"182442818
1422418")
470 CALL CHAR(118,"181818C3
C3C7E7E")
480 CALL CHAR(119,"7E7E7E7E7
```

```
E7E7EFF")
490 CALL CHAR(126,"FFFEFCF8F
0E0C080")
500 CALL CHAR(123,"FF7F3F1F0
F070301")
510 CALL CHAR(122,"FFFFFFFFF
FFFFFFF")
520 FOR I=1 TO 22
530 READ ZE,SP,CH,WH
540 CALL HCHAR(ZE,SP,CH,WH)
550 NEXT I
560 DATA 1,13,111,1,1,16,111
,1,2,13,118,1,2,16,118,1,3,1
1,124,1,3,13,119,1,3,16,119,
1,3,18,125,1
570 DATA 4,12,122,6
580 DATA 5,12,126,1,5,14,123
,1,5,15,126,1,5,17,123,1,6,1
2,125,1,6,14,124,1,6,15,125,
1,6,17,124,1
590 DATA 7,12,122,6
600 DATA 8,11,123,1,8,14,123
,1,8,15,126,1,8,18,126,1
610 XSCH=11
620 SPIEL=0
630 RSCH=5
640 POS1=15
650 CALL SCREEN(2)
660 CALL COLOR(2,12,2)
670 CALL CHAR(100,"183C7EFFF
F0000FF")
680 CALL CHAR(128,"001818181
8181800")
690 CALL CHAR(129,"181818181
8187EFF")
700 CALL CHAR(136,"8181423C3
C3C3C18")
710 CALL CHAR(137,"2424A57E5
A5A7E99")
720 CALL COLOR(13,12,2)
730 CALL COLOR(14,11,2)
740 CALL COLOR(9,14,2)
750 CALL COLOR(10,10,2)
760 CALL COLOR(11,7,2)
770 CALL COLOR(12,6,2)
780 CALL HCHAR(21,1,100,30)
790 CALL HCHAR(20,POS1,129)
800 CALL HCHAR(22,5,RSCH+48)
810 FOR SCH=9 TO 1 STEP -1
820 CALL HCHAR(22,27,SCH+48)
830 SPIEL=SPIEL+1
840 HPOS=XSCH
850 HPOS2=INT(RND*(2*SCH+XSC
H-4))+1
860 CALL HCHAR(HPOS-1,HPOS2,
137)
870 CALL HCHAR(HPOS,HPOS2,13
6)
880 HPOS=HPOS+1
890 IF HPOS=20 THEN 1460
900 FOR I=1 TO SCH
910 CALL KEY(1,K,S)
920 CALL JOYST(1,X,Y)
930 IF (S=0)*(X=0)THEN 970
940 GOTO 1200
950 CALL HCHAR(20,POS1,129)
960 CALL SOUND(-50,440,5)
970 NEXT I
980 CALL HCHAR(HPOS-1,HPOS2,
127)
990 GOTO 870
1000 IF SPIEL/ANZ<>INT(SPIEL
/ANZ)THEN 830
1010 NEXT SCH
1020 RSCH=RSCH+XSCH-10
1030 IF RSCH<10 THEN 1050
1040 RSCH=9
1050 XSCH=XSCH+1
1060 CALL HCHAR(22,5,RSCH+48
)
1070 PUNKT=PUNKT+100*XSCH
1080 CALL SOUND(300,524,1)
1090 CALL SOUND(300,588,1)
1100 CALL SOUND(600,660,1)
1110 PUNKT=PUNKT+100*XSCH
1120 CALL SOUND(225,698,3,88
0,1)
1130 CALL SOUND(75,784,1)
1140 CALL SOUND(150,880,1)
1150 CALL SOUND(150,988,2,78
2,1)
1160 CALL SOUND(300,1046,5,7
82,4,658,0)
1170 IF XSCH>16 THEN 1690
1180 CALL COLOR(14,XSCH,2)
1190 GOTO 810
1200 IF X<>-4 THEN 1250
1210 IF POS1-1<1 THEN 1240
1220 CALL HCHAR(20,POS1,127)
1230 POS1=POS1-1
1240 GOTO 950
1250 IF X<>4 THEN 1300
1260 IF POS1+1>30 THEN 1290
1270 CALL HCHAR(20,POS1,127)
1280 POS1=POS1+1
1290 GOTO 950
1300 IF K<>18 THEN 970
1310 FOR J=19 TO HPOS STEP -
1
1320 CALL HCHAR(J,POS1,128)
1330 NEXT J
1340 IF HPOS2<>POS1 THEN 143
0
1350 CALL SOUND(-200,1200,0)
1360 PUNKT=PUNKT+(21-HPOS)*(
11-SCH)
1370 HILF$=STR$(PUNKT)
1380 FOR J=1 TO LEN(HILF$)
1390 CALL HCHAR(22,15+J,ASC(
SEG$(HILF$,J,1)))
1400 NEXT J
```

```
1410 CALL VCHAR(10,POS1,127,
10)
1420 GOTO 1000
1430 CALL VCHAR(HPOS,POS1,12
7,20-HPOS)
1440 CALL SOUND(-200,-2,0)
1450 GOTO 950
1460 RSCH=RSCH-1
1470 CALL SCREEN(10)
1480 CALL SOUND(-1000,-6,0)
1490 CALL HCHAR(10,1,127,352
)
1500 CALL SCREEN(2)
1510 IF RSCH<0 THEN 1550
1520 CALL HCHAR(22,5,RSCH+48
)
1530 CALL HCHAR(20,POS1,129)
1540 GOTO 1000
1550 CALL CLEAR
1560 FOR I=5 TO 8
1570 CALL COLOR(I,16,2)
1580 NEXT I
1590 PRINT "SIE HABEN LEIDER
 VERLOREN":"ABER PROBIEREN S
IE ES DOCH  RUHIG NOCH EINMA
L."
1600 PRINT "SIE HATTEN ";PUN
KT;" PUNKTE":"UND HATTEN SCH
WIERIGKEITS-  GRAD :";SCH
1610 PRINT "IN STUFE:";XSCH-
11
1620 PRINT "NOCHMAL?"
1630 CALL KEY(0,K,S)
1640 IF S=0 THEN 1630
1650 IF K<>74 THEN 1670
1660 GOTO 100
1670 IF K<>78 THEN 1630
1680 END
1690 CALL CLEAR
1700 FOR I=1 TO 14
1710 CALL COLOR(I,2,16)
1720 NEXT I
1730 CALL SCREEN(16)
1740 PRINT "BRAVO!!": :"ABSO
LUTE SPITZENKLASSE"
1750 PRINT "PUNKTE:";PUNKT:"
SIE HABEN ES BIS ZUM LETZTEN
SCHWIERIGKEITSGRAD DURCHGE-
HALTEN!"
1760 PRINT "NOCHMAL?"
1770 CALL KEY(0,K,S)
1780 IF S=0 THEN 1770
1790 IF K=74 THEN 100
1800 IF K<>78 THEN 1770
1810 END
```

Die nächste TI REVUE erscheint am 28. Oktober

```
10 ! **********************
11 ! *                    *
12 ! * ROUTINE ZUM AUS-   *
13 ! *   DRUCKEN VON      *
14 ! * DFUE-PROTOKOLLEN   *
15 ! *   Copyright by     *
16 ! *    Dieter Orth     *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *  Terminal-Emulator *
23 ! *         oder       *
24 ! * RTE II und Disk-Drive*
25 ! *       Drucker      *
26 ! *  Speicherbelegung  *
27 ! *     2754 Bytes     *
28 ! *                    *
29 ! **********************
```

```
100 CALL CLEAR :: CALL SCREE
N(2)
110 DISPLAY AT(1,1):"*******
**********************" :: D
ISPLAY AT(2,1):"*" :: DISPLA
Y AT(2,28):"*"
120 DISPLAY AT(3,1):"*......
.LESEPROGRAMM.......*" :: DI
SPLAY AT(4,1):"*" :: DISPLAY
 AT(4,28):"*"
130 DISPLAY AT(5,1):"*...FUE
R DFUE-TEXTE VON....*" :: DI
SPLAY AT(6,1):"*" :: DISPLAY
 AT(6,28):"*"
140 DISPLAY AT(7,1):"*......
.'DISKETTEN'........*" :: DI
SPLAY AT(8,1):"*" :: DISPLAY
 AT(8,28):"*"
150 DISPLAY AT(9,1):"*......
..IM FORMAT.........*" :: DI
SPLAY AT(10,1):"*" :: DISPLA
Y AT(10,28):"*" :: DISPLAY A
T(11,1):"*  DISPLAY/VARIABLE
 80/128 *" :: DISPLAY AT(11,
28):"*"
160 DISPLAY AT(12,1):"*" ::
DISPLAY AT(12,28):"*" :: DIS
PLAY AT(13,1):"*" :: DISPLAY
 AT(13,28):"*" :: DISPLAY AT
(14,1):"*" :: DISPLAY AT(14,
28):"*"
170 DISPLAY AT(15,1):"*.....
...VERSION 1.2 ......*" :: D
ISPLAY AT(16,1):"*" :: DISPL
AY AT(16,28):"*" :: DISPLAY
AT(17,1):"******************
**********"
180 CALL SCREEN(4):: FOR I=1
 TO 750 :: NEXT I
```

```
190 !-----------------------
200 ! ROUTINE ZUM AUSDRUCKEN
      DER DFUE-PROTOKOLLE
      VON DISKETTEN
210 ! LITERATUR:
      TI REVUE 2/84,S.44
220 ! CR BY DIETER ORTH
            HARTGERSTR. 12
          33 BRAUNSCHWEIG
230 !-----------------------
     MENUE
     -----------------------
240 CALL CLEAR :: DISPLAY AT
(2,4):"<GERAETE.DATEINAMEN>"
250 !-----------------------
260 DISPLAY AT(4,4):"===>" :
: ACCEPT AT(4,8)SIZE(15)VALI
DATE(UALPHA,DIGIT,"./-_"):FI
LE$ :: IF FILE$="" THEN 260
270 !-----------------------
280 DISPLAY AT(7,4):"<TERMIN
AL-EMULATOR>" :: DISPLAY AT(
8,4):"--------------------"
290 DISPLAY AT(9,4):"TE II
 =<1>" :: DISPLAY AT(10,17):
"> <" :: DISPLAY AT(11,4):"T
E 1200 =<2>" :: ACCEPT AT(10
,18)SIZE(1)VALIDATE("12"):T$
300 IF T$="" THEN 290 :: IF
T$="1" THEN 320 :: IF T$="2"
 THEN 330 ELSE 290
310 DISPLAY AT(24,1):FILE$
320 OPEN #1:FILE$,INPUT ,DIS
PLAY ,VARIABLE 80 :: GOTO 34
0
330 OPEN #1:FILE$,INPUT ,DIS
PLAY ,VARIABLE 128
340 DISPLAY AT(14,4):"DRUCKE
R  J/N > <" :: ACCEPT AT(14,
18)SIZE(1)VALIDATE("JN "):C$
350 IF C$="" THEN 340 :: IF
C$="J" THEN 380 :: IF C$="N"
 THEN 500
360 DISPLAY AT(24,1):FILE$
370 !-----------------------
     DRUCKER-AUSGABE
     -----------------------
380 DISPLAY AT(17,4):"RS232.
...<1>" :: DISPLAY AT(18,17)
:"> <" :: DISPLAY AT(19,4):"
SELEKTIV <2>" :: ACCEPT AT(1
8,18)SIZE(1)VALIDATE("12"):D
$
390 IF D$="" THEN 380 :: IF
D$="1" THEN 400 :: IF D$="2"
 THEN 420
400 OPEN #2:"RS232.BA=9600.D
A=8.PA=E",OUTPUT,DISPLAY ::
GOTO 450
410 !-----------------------
420 DISPLAY AT(21,4):"DRUCKE
RNAME:" :: ACCEPT AT(22,4)SI
ZE(28)VALIDATE(UALPHA,DIGIT,
".=,"):P$ :: IF P$="" THEN 4
20
430 OPEN #2:P$,OUTPUT,DISPLA
Y
440 !-----------------------
450 LINPUT #1:A$
460 IF EOF(1)THEN 570
470 PRINT #2:A$
480 GOTO 450
490 !-----------------------
     BILDSCHIRM-AUSGABE
     -----------------------
500 CALL CLEAR :: DISPLAY AT
(21,1):"BITTE DIE SPACE - TA
STE.....BENUTZEN!" :: FOR I=
1 TO 750 :: NEXT I :: CALL C
LEAR
510 LINPUT #1:A$
520 IF EOF(1)THEN 570
530 PRINT A$ :: FOR I=0 TO 1
540 CALL KEY(0,K,S):: IF S=0
 THEN 540 :: IF K=32 THEN 55
0 :: GOTO 540
550 NEXT I :: GOTO 510
560 !-----------------------
     MENUE
     -----------------------
570 DISPLAY AT(24,4):"WEITER
...J/N > <" :: ACCEPT AT(24,
18)SIZE(1)VALIDATE("JN"):S$
:: IF S$="" THEN 570 :: IF S
$="J" THEN RUN 240 :: IF S$=
"N" THEN 580
580 RUN "DSK1.LOAD" :: REM--
--->RUN "DSK1.MENUE"
590 END
```

Mehr Geld verdienen . . .

Wenn Sie den Wunsch zu selbständiger Arbeit haben und gut verdienen wollen –

HIER IST DIE LÖSUNG:

Werden Sie Partner von **Michael Meister, EDV-Marketing**
Rheinstr. 47, 7500 Karlsruhe 21, Auftragsdienst Frau Staack, Tel. 07 21/55 46 01

Keine Broschüren, keine Konzepte — sondern reelle Chancen — geringes Startkapital erforderlich, ab DM 100,—. Technisches Interesse und kaufmännische Kenntnisse von Vorteil! Bitte benutzen Sie die Kontaktkarte und fügen Sie einen V-Scheck von DM 5,— bei! (Schutzgebühr)

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE

KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre TI REVUE nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Euer" Kiosk nicht beliefert wurde? Kein Problem! Für ganze 30 DM liefern wir Euch per Post sechs Hefte ins Haus (Ausland 40 DM). Einfach den Bestellschein auf der nächsten Seite ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). TI REVUE kommt dann pünktlich ins Haus.

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen– Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für sechs Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

DAS SUPER-SONDER-ANGEBOT: PRIVATE KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen ab sofort die TI-Revue: KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!

Wir veröffentlichen nur Kleinanzeigen privater Inserenten, keine gewerblichen Anzeigen. Die kosten pro Millimeter DM 2,50 plus Mehrwertsteuer!

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!

Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!

Die Insertion ist nicht vom Kauf des Heftes abhängig!

Wir behalten uns vor, Anzeigen, die nicht zum Themenkreis des Heftes – Computer – gehören, nicht abzudrucken oder sie nur insoweit zu berücksichtigen, wie es der Umfang des kostenlosen Anzeitenteils zuläßt.

KASSETTEN

Selbstverständlich gibt es in diesem Heft wieder einen Kassetten-Service. Dabei können wir Ihnen allerdings die Qual der Wahl nicht ersparen. Denn: In Ihrem Interesse haben wir dieses Special sozusagen „geviertelt".

Viertel 1: Alle Utilities und Anwender-Programme in TI Basic
Viertel 2: Alle Spiele in TI Basic
Viertel 3: Alle Utilities/Anwenderprogramme in Extended Basic
Viertel 4: Alle Spiele in Extended Basic.

Und weil es uns Spaß macht, haben wir noch ein fünftes Viertel angehängt: den Disketten-Service. Da sind dann allerdings alle Programme drauf.

Und nun zu den harten, nackten Geschäftsbedingungen.

Sie können zahlen:
Gegen Bankabbuchung am Versandtag
Gegen Vorausrechnung, wobei am Tag des Geldeinganges der Versand erfolgt.

Sie können kombinieren:
Jedwedes Programmpaket miteinander — etwa 1 + 4 oder 1,2,3 oder nur 4 und so weiter.

ANZEIGENSERVICE

Die große Börse für jeden Zweck in der TI REVUE. Kostenlos für Privat-Inserenten. Spott-billig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Achtung! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt nicht TI-typisch ist oder gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatanbieter: maximal acht Zeilen à 28 Anschläge.

Name ______
Vorname ______
Str.Nr. ______
(PLZ) Ort ______

TI REVUE
Anzeigenabteilg.
Postfach 1107
8044 Lohhof

ABO SERVICE-KARTE

6
TI

Coupon

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten sechs Ausgaben an untenstehende Anschrift.

Name ______
Vorname ______
Straße/Hausnr. ______
Plz/Ort ______

Ich bezahle:
☐ per beiliegendem Verrechnungsscheck
☐ gegen Rechnung
☐ bargeldlos per Bankeinzug von meinem Konto

bei (Bank) und Ort ______
Kontonummer ______
Bankleitzahl ______
(steht auf jedem Kontoauszug)
Unterschrift ______

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift ______

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

TI REVUE
Abo-Service SP2
Postfach 1107
8044 UNTERSCHLEISSHEIM

SERVICE

Das sind die Preise: Pro einzelnes Programm-Paket DM 20,–
Für zwei Programmpakete DM 35,– (5,– DM Ersparnis)
Für drei Programmpakete DM 50,– (10,– DM Ersparnis)
Für alle vier Programmpakete DM 60,– (20,– DM Ersparnis)
(Jedes Programmpaket auf Kassette!)
Für die komplette Diskette – alle vier Programm-Pakete – DM 75,– (hier ist keine Zusammenstellung möglich!)

Die Inhalte der einzelnen Programm-Pakete ersehen Sie aus nachfolgender Übersicht, die Komplett-Diskette enthält alle vier Programm-Pakete!

ACHTUNG: KEIN GELD EINSENDEN! NUR DEN BERECHTIGUNGSCOUPON! WIR MELDEN UNS, WENN DIE WARE VERSANDBEREIT IST!!! SELBSTVERSTÄNDLICH VOLLES UMTAUSCHRECHT BEI NICHTFUNKTIONIEREN.

Programm-Paket 1:
Alle Spiele in TI-Basic
wie Erdbeer-Paul, Apfelfänger, Seeschlacht, Space Invaders, Castle of Dom, Catch with Car, Sky-Driver, TI-Mühle, Irvel.

Programmpaket 2:
Alle Spiele in Ext.-Basic
wie Space 2000, Tore schießen, TI-Fall, Spickzettel, Adventage, Ecarte, Boogabo.

Programmpaket 3:
Alle Anwenderprogramme, Utilities usw. in Basic
wie Dreiecks- und Kreisberechnung, Kompositionen, Dissassembler, Copyright-Zeichen.

Programmpaket 4:
Alle Anwenderprogramme, Utilities usw. in Ext. Basic wie Hexdump, Supergrafik, Toch Typing, Sprite-Set DFÜ, Tabellen-Erstellung, Steuerberechnung, Prognose, Karnoug Veitch, Reset, 3x3-Matrix, Softplot oder Pokelisten.

ACHTUNG! Wir weisen ausdrücklich darauf hin, daß das Spiel „Irvel" wegen Überlänge nur in der Ursprungsversion des Autors kopiert wurde. Eventuell müssen Sie daher dieses Programm gemäß unserem Listing überprüfen.

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen
Programmsammlung 1 ( ) Programmsammlung 2 ( )
Programmsammlung 3 ( ) Programmsammlung 4 ( )
die Komplett-Diskette ( )
Zutreffendes bitte ankreuzen!

Ich zahle:

Gegen Vorausrechnung ( ) – Versand am Tag des Geldeinganges
Gegen Bankabbuchung am Versandtag ( )
Meine Bank (mit Ortsname) .....
Meine Kontonummer .....
Meine Bankleitzahl ..... (steht auf jedem Bankauszug)
Vorname ..... Nachname .....
Str./Nr. ..... PLZ / Ort .....
Hiermit bestätigen ich mit meiner Unterschrift, Ihre Verkaufsbedingungen gelesen zu haben und zu akzeptieren.

Unterschrift .....

Bitte ausschneiden und einsenden an
TI-SPECIAL 2
KASSETTENSERVICE
Postfach 1107
8044 Unterschleißheim

COMPUTER AKTUELL
Die deutsche Computer-Illustrierte

Nr. 9/10 Sept / Okt - DM 9,80/ÖS 80 /SFR 9,80 B 20131 F

GEWINNEN SIE EINEN COMPUTER

Alles, was Sie über Drucker wissen wollen!

NEU!

Das große Drucker-Special

IM TEST:

16 Drucker von Philips über Mannesmann Comdata, Brother, Star, Epson, Speedy, Fujitsu, Honeywell, Juki, Microscan und Sharp bis Sony

MARKTÜBER-SICHT:

Alle Drucker Printer/Plotter Von Thermo bis Typenrad

DAS AKTUELLE COMPUTER-MAGAZIN

DAS BLATT, DAS KEINS VOR DEN MUND NIMMT

JETZT AN IHREM KIOSK

```
10 ! **********************
11 ! *                    *
12 ! *  KARNOUGH-VEITCH   *
13 ! *      DIAGRAMM      *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *   Bernard Schmidt  *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! * TI99/4A  Konsole   *
21 ! *     Ext. Basic     *
22 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     11702 Bytes    *
28 ! *                    *
29 ! **********************
30 !
260 CALL CLEAR
270 DIM C(16),R$(62)
280 !
290 ! ZEICHENDEFINITIONEN
300 DATA "7C0000384447C4444",
"7C000007824382478","7C00003C
4040403C","7C000007824242478"
310 DATA "0101010101010101",
"00000000000000FF","0101010101
0101FF","FF0000000000000000"
,"3080808080803080"
320 RESTORE 300 :: FOR I=1 T
O 4 :: READ A$ :: CALL CHAR(
100+I,A$):: NEXT I
330 RESTORE 310 :: FOR I=1 T
O 5 :: READ A$ :: CALL CHAR(
32+I,A$):: NEXT I
340 !
350 !DATEN FUER WAHRHEITSTAB
ELLE
360 !FUER 0
370 DATA 5,5,5,7,5,9,5,11,6,
5,6,7,6,9,7,5,7,7,7,11,8,5,8
,7,9,5,9,9,9,11
380 DATA 10,5,10,9,11,5,11,1
1,12,5,13,7,13,9,13,11,14,7,
14,9,15,7,15,11,16,7,17,9,17
,11
390 DATA 18,9,19,11
400 !FUER 1
410 DATA 6,11,7,9,8,9,8,11,9
,7,10,7,10,11,11,7,11,9,12,7
,12,9,12,11,13,5,14,5,14,11
420 DATA 15,5,15,9,16,5,16,9
,16,11,17,5,17,7,18,5,18,7,1
8,11,19,5,19,7,19,9,20,5,20,
7,20,9,20,11
430 !
440 !ERGEBNISDATEN
450 DATA 0,1,f,d,b,e,c,a,h,g
,ef,ed,eb,eh,fc,cd,bc,ch,af,
ad,ab,ah,fg,gd,bg,gh,eg
460 DATA ec,ac,ag,fh,fd,bd,b
h,efg,egd,ebg,efc,ecd,ebc,af
c,acd,abc,afg,agd,abg,egh
470 DATA ech,ach,agh,efh,efd
,ebd,ebh,fch,fcd,bcd,bch,afh
,afd,abd,abh,fgh,fgd,bgd,bgh
480 DATA efgh,efgd,efch,efcd
,ebgh,ebgd,ebch,ebcd,afgh,af
gd,afch,afcd,abgh,abgd,abch,
abcd
490 !DARSTELLUNG WAHRHEITSTA
BELLE
500 A$="WAHRHEITSTABELLE" ::
 FOR I=1 TO 16 :: CALL HCHAR
(1,2+I,ASC(SEG$(A$,I,1))):: 
NEXT I
510 FOR I=4 TO 12 STEP 2 ::
CALL HCHAR(4,I,35):: CALL HC
HAR(3,I,33):: NEXT I
520 FOR I=3 TO 13 STEP 2 ::
CALL HCHAR(4,I,34):: NEXT I
530 FOR I=4 TO 12 STEP 2 ::
FOR J=5 TO 20 :: CALL HCHAR(
J,I,33):: NEXT J :: NEXT I
540 FOR I=1 TO 4 :: J=2*I+3
:: CALL HCHAR(3,J,96+I):: NE
XT I
550 CALL HCHAR(3,13,65)
560 FOR I=1 TO 10 :: CALL HC
HAR(4+I,3,47+I):: NEXT I
570 FOR I=1 TO 6 :: CALL HCH
AR(14+I,2,49):: CALL HCHAR(1
4+I,3,47+I):: NEXT I
580 RESTORE 370 :: FOR I=1 T
O 32 :: READ A,B :: CALL HCH
AR(A,B,48):: NEXT I
590 RESTORE 410 :: FOR I=1 T
O 32 :: READ A,B :: CALL HCH
AR(A,B,49):: NEXT I
600 A$="KORREKTUR  ->K" :: F
OR I=1 TO 14 :: CALL HCHAR(3
,17+I,ASC(SEG$(A$,I,1))):: N
EXT I
610 A$="1=HIGH" :: FOR I=1 T
O 6 :: CALL HCHAR(6,17+I,ASC
(SEG$(A$,I,1))):: NEXT I
620 A$="0=LOW" :: FOR I=1 TO
 5 :: CALL HCHAR(8,17+I,ASC(
SEG$(A$,I,1))):: NEXT I
630 A$="X=DON'T CARE" :: FOR
 I=1 TO 12 :: CALL HCHAR(10,
17+I,ASC(SEG$(A$,I,1))):: NE
XT I
640 A$="BITTE A EINGEBEN :"
:: FOR I=1 TO 18 :: CALL HCH
AR(22,3+I,ASC(SEG$(A$,I,1)))
:: NEXT I
650 !
660 !WERTE FUER A EINGEBEN
670 FOR I=0 TO 15
680 CALL KEY(0,K,S):: IF S<=
0 THEN 680 :: IF K=48 OR K=4
```

```
9 OR K=88 THEN 690 ELSE IF K
=75 THEN 700 ELSE 680
690 CALL HCHAR(5+I,13,K):: G
OTO 710
700 IF I=0 THEN GOTO 680 ELS
E I=I-1 :: GOTO 680
710 NEXT I
720 CALL HCHAR(22,3,32,19)::
 CALL HCHAR(3,17,32,15):: FO
R I=1 TO 5 STEP 2 :: CALL HC
HAR(5+I,17,32,13):: NEXT I
730 !
740 !KV-DIAGRAMM ERSTELLEN
750 A$="KV-DIAGRAMM" :: FOR
I=1 TO 11 :: CALL HCHAR(3,17
+I,ASC(SEG$(A$,I,1)))):: NEXT
 I
760 CALL HCHAR(7,20,34,4)::
CALL VCHAR(8,19,33,4):: CALL
 VCHAR(8,24,37,4):: CALL HCH
AR(12,20,36,4)
770 CALL HCHAR(6,21,36,2)::
CALL HCHAR(5,21,100):: CALL
VCHAR(9,25,33,2):: CALL VCHA
R(9,26,99)
780 CALL VCHAR(10,18,37,2)::
 CALL VCHAR(10,17,97):: CALL
 HCHAR(13,22,34,2):: CALL HC
HAR(14,22,98)
790 FOR I=0 TO 15 :: CALL GC
HAR(5+I,13,C(I)):: NEXT I
800 CALL HCHAR(8,20,C(0))::
CALL HCHAR(8,21,C(1)):: CALL
 HCHAR(9,20,C(2)):: CALL HCH
AR(9,21,C(3))
810 CALL HCHAR(8,23,C(4))::
CALL HCHAR(8,22,C(5)):: CALL
 HCHAR(9,23,C(6)):: CALL HCH
AR(9,22,C(7))
820 CALL HCHAR(11,20,C(8))::
 CALL HCHAR(11,21,C(9)):: CA
LL HCHAR(10,20,C(10)):: CALL
 HCHAR(10,21,C(11))
830 CALL HCHAR(11,23,C(12))
: CALL HCHAR(11,22,C(13))::
CALL HCHAR(10,23,C(14)):: CA
LL HCHAR(10,22,C(15))
840 !
850 !AUSWERTUNG
860 FOR I=0 TO 15 :: C(I)=C(
I)-48 :: NEXT I
870 RESTORE 450
880 FOR I=0 TO 81 :: READ R$
(I):: NEXT I :: ERG$=""
890 !
900 !16er BLOCK
910 A=0 :: FOR I=0 TO 15 ::
A=A+C(I):: NEXT I
920 IF A=0 THEN ERG$=R$(0)EL
SE 930 :: GOTO 2350
930 FOR I=0 TO 15 :: IF C(I)
=0 THEN 960
940 NEXT I :: ERG$=R$(1):: G
OTO 2350
950 !
960 !8er BLOCK
970 IF C(0)>=1 AND C(1)>=1 A
ND C(2)>=1 AND C(3)>=1 AND C
(10)>=1 AND C(11)>=1 AND C(8
)>=1 AND C(9)>=1 THEN ERG$=E
RG$&R$(2)&"+" ELSE 990
980 FOR I=0 TO 3 :: FOR J=8
TO 11 :: C(I),C(J)=0 :: NEXT
 J :: NEXT I
990 IF C(1)>=1 AND C(5)>=1 A
ND C(3)>=1 AND C(7)>=1 AND C
(11)>=1 AND C(15)>=1 AND C(9
)>=1 AND C(13)>=1 THEN ERG$=
ERG$&R$(3)&"+" ELSE 1010
1000 C(1),C(5),C(3),C(7),C(1
1),C(15),C(9),C(13)=0
1010 IF C(5)>=1 AND C(7)>=1
AND C(15)>=1 AND C(13)>=1 AN
D C(4)>=1 AND C(6)>=1 AND C(
14)>=1 AND C(12)>=1 THEN ERG
$=ERG$&R$(4)&"+" ELSE 1030
1020 C(5),C(7),C(15),C(13),C
(4),C(6),C(14),C(12)=0
1030 IF C(0)>=1 AND C(1)>=1
AND C(2)>=1 AND C(3)>=1 AND
C(5)>=1 AND C(6)>=1 AND C(7)
>=1 AND C(4)>=1 THEN ERG$=ER
G$&R$(5)&"+" ELSE 1050
1040 FOR I=0 TO 3 :: FOR J=4
 TO 7 :: C(I),C(J)=0 :: NEXT
 J :: NEXT I
1050 IF C(2)>=1 AND C(3)>=1
AND C(7)>=1 AND C(6)>=1 AND
C(10)>=1 AND C(11)>=1 AND C(
15)>=1 AND C(14)>=1 THEN ERG
$=ERG$&R$(6)&"+" ELSE 1070
1060 C(2),C(3),C(7),C(6),C(1
0),C(11),C(15),C(14)=0
1070 IF C(8)>=1 AND C(9)>=1
AND C(10)>=1 AND C(11)>=1 AN
D C(12)>=1 AND C(13)>=1 AND
C(14)>=1 AND C(15)>=1 THEN E
RG$=ERG$&R$(7)&"+" ELSE 1090
1080 FOR I=8 TO 15 :: C(I)=0
 :: NEXT I
1090 IF C(0)>=1 AND C(2)>=1
AND C(4)>=1 AND C(6)>=1 AND
C(8)>=1 AND C(10)>=1 AND C(1
2)>=1 AND C(14)>=1 THEN ERG$
=ERG$&R$(8)&"+" ELSE 1110
1100 FOR I=0 TO 14 STEP 2 ::
 C(I)=0 :: NEXT I
1110 IF C(0)>=1 AND C(1)>=1
AND C(5)>=1 AND C(4)>=1 AND
C(8)>=1 AND C(9)>=1 AND C(13
)>=1 AND C(12)>=1 THEN ERG$=
```

```
ERG$&R$(9)&"+" ELSE 1140
1120 C(0),C(1),C(5),C(4),C(8
),C(9),C(13),C(12)=0
1130 !
1140 !4er BLOCK
1150 IF C(0)>=1 AND C(1)>=1
AND C(2)>=1 AND C(3)>=1 THEN
 ERG$=ERG$&R$(10)&"+" ELSE 1
170
1160 FOR I=0 TO 3 :: C(I)=0
:: NEXT I
1170 IF C(1)>=1 AND C(3)>=1
AND C(5)>=1 AND C(7)>=1 THEN
 ERG$=ERG$&R$(11)&"+" ELSE 1
190
1180 FOR I=1 TO 7 STEP 2 ::
C(I)=0 :: NEXT I
1190 IF C(4)>=1 AND C(5)>=1
AND C(6)>=1 AND C(7)>=1 THEN
 ERG$=ERG$&R$(12)&"+" ELSE 1
210
1200 FOR I=4 TO 7 :: C(I)=0
:: NEXT I
1210 IF C(0)>=1 AND C(2)>=1
AND C(4)>=1 AND C(6)>=1 THEN
 ERG$=ERG$&R$(13)&"+" ELSE 1
230
1220 FOR I=0 TO 6 STEP 2 ::
C(I)=0 :: NEXT I
1230 IF C(2)>=1 AND C(3)>=1
AND C(10)>=1 AND C(11)>=1 TH
EN ERG$=ERG$&R$(14)&"+" ELSE
 1250
1240 C(2),C(3),C(10),C(11)=0
1250 IF C(3)>=1 AND C(7)>=1
AND C(11)>=1 AND C(15)>=1 TH
EN ERG$=ERG$&R$(15)&"+" ELSE
 1270
1260 C(3),C(7),C(11),C(15)=0
1270 IF C(7)>=1 AND C(6)>=1
AND C(15)>=1 AND C(14)>=1 TH
EN ERG$=ERG$&R$(16)&"+" ELSE
 1290
1280 C(7),C(6),C(15),C(14)=0
1290 IF C(2)>=1 AND C(10)>=1
 AND C(6)>=1 AND C(14)>=1 TH
EN ERG$=ERG$&R$(17)&"+" ELSE
 1310
1300 C(2),C(10),C(6),C(14)=0
1310 IF C(8)>=1 AND C(9)>=1
AND C(10)>=1 AND C(11)>=1 TH
EN ERG$=ERG$&R$(18)&"+" ELSE
 1330
1320 FOR I=8 TO 11 :: C(I)=0
 :: NEXT I
1330 IF C(9)>=1 AND C(11)>=1
 AND C(13)>=1 AND C(15)>=1 T
HEN ERG$=ERG$&R$(19)&"+" ELS
E 1350
1340 FOR I=9 TO 15 STEP 2 ::
 C(I)=0 :: NEXT I
1350 IF C(12)>=1 AND C(13)>=
1 AND C(14)>=1 AND C(15)>=1
THEN ERG$=ERG$&R$(20)&"+" EL
SE 1370
1360 FOR I=12 TO 15 :: C(I)=
0 :: NEXT I
1370 IF C(8)>=1 AND C(10)>=1
 AND C(12)>=1 AND C(14)>=1 T
HEN ERG$=ERG$&R$(21)&"+" ELS
E 1390
1380 FOR I=8 TO 14 STEP 2 ::
 C(I)=0 :: NEXT I
1390 IF C(0)>=1 AND C(1)>=1
AND C(8)>=1 AND C(9)>=1 THEN
 ERG$=ERG$&R$(22)&"+" ELSE 1
410
1400 C(0),C(1),C(8),C(9)=0
1410 IF C(1)>=1 AND C(5)>=1
AND C(9)>=1 AND C(13)>=1 THE
N ERG$=ERG$&R$(23)&"+" ELSE
1430
1420 C(1),C(5),C(9),C(13)=0
1430 IF C(5)>=1 AND C(4)>=1
AND C(13)>=1 AND C(12)>=1 TH
EN ERG$=ERG$&R$(24)&"+" ELSE
 1450
1440 C(5),C(4),C(13),C(12)=0
1450 IF C(0)>=1 AND C(4)>=1
AND C(8)>=1 AND C(12)>=1 THE
N ERG$=ERG$&R$(25)&"+" ELSE
1470
1460 C(0),C(4),C(8),C(12)=0
1470 IF C(0)>=1 AND C(1)>=1
AND C(5)>=1 AND C(4)>=1 THEN
 ERG$=ERG$&R$(26)&"+" ELSE 1
490
1480 C(0),C(1),C(5),C(4)=0
1490 IF C(2)>=1 AND C(3)>=1
AND C(7)>=1 AND C(6)>=1 THEN
 ERG$=ERG$&R$(27)&"+" ELSE 1
510
1500 C(2),C(3),C(7),C(6)=0
1510 IF C(10)>=1 AND C(11)>=
1 AND C(15)>=1 AND C(14)>=1
THEN ERG$=ERG$&R$(28)&"+" EL
SE 1530
1520 C(10),C(11),C(15),C(14)
=0
1530 IF C(8)>=1 AND C(9)>=1
AND C(13)>=1 AND C(12)>=1 TH
EN ERG$=ERG$&R$(29)&"+" ELSE
 1550
1540 C(8),C(9),C(13),C(12)=0
1550 IF C(0)>=1 AND C(2)>=1
AND C(10)>=1 AND C(8)>=1 THE
N ERG$=ERG$&R$(30)&"+" ELSE
1570
1560 C(0),C(2),C(10),C(8)=0
1570 IF C(1)>=1 AND C(3)>=1
AND C(11)>=1 AND C(9)>=1 THE
```

```
N ERG$=ERG$&R$(31)&"+" ELSE
1590
1580 C(1),C(3),C(11),C(9)=0
1590 IF C(5)>=1 AND C(7)>=1
AND C(15)>=1 AND C(13)>=1 TH
EN ERG$=ERG$&R$(32)&"+" ELSE
 1610
1600 C(5),C(7),C(15),C(13)=0
1610 IF C(4)>=1 AND C(6)>=1
AND C(14)>=1 AND C(12)>=1 TH
EN ERG$=ERG$&R$(33)&"+" ELSE
 1640
1620 C(4),C(6),C(14),C(12)=0
1630 !
1640 !2er BLOCK
1650 IF C(0)>=1 AND C(1)>=1
THEN ERG$=ERG$&R$(34)&"+" EL
SE 1670
1660 C(0),C(1)=0
1670 IF C(1)>=1 AND C(5)>=1
THEN ERG$=ERG$&R$(35)&"+" EL
SE 1690
1680 C(1),C(5)=0
1690 IF C(5)>=1 AND C(4)>=1
THEN ERG$=ERG$&R$(36)&"+" EL
SE 1710
1700 C(5),C(4)=0
1710 IF C(2)>=1 AND C(3)>=1
THEN ERG$=ERG$&R$(37)&"+" EL
SE 1730
1720 C(2),C(3)=0
1730 IF C(3)>=1 AND C(7)>=1
THEN ERG$=ERG$&R$(38)&"+" EL
SE 1750
1740 C(3),C(7)=0
1750 IF C(6)>=1 AND C(7)>=1
THEN ERG$=ERG$&R$(39)&"+" EL
SE 1770
1760 C(6),C(7)=0
1770 IF C(10)>=1 AND C(11)>=
1 THEN ERG$=ERG$&R$(40)&"+"
ELSE 1790
1780 C(10),C(11)=0
1790 IF C(11)>=1 AND C(15)>=
1 THEN ERG$=ERG$&R$(41)&"+"
ELSE 1810
1800 C(11),C(15)=0
1810 IF C(15)>=1 AND C(14)>=
1 THEN ERG$=ERG$&R$(42)&"+"
ELSE 1830
1820 C(15),C(14)=0
1830 IF C(8)>=1 AND C(9)>=1
THEN ERG$=ERG$&R$(43)&"+" EL
SE 1850
1840 C(8),C(9)=0
1850 IF C(9)>=1 AND C(13)>=1
 THEN ERG$=ERG$&R$(44)&"+" E
LSE 1870
1860 C(9),C(13)=0
1870 IF C(13)>=1 AND C(12)>=
1 THEN ERG$=ERG$&R$(45)&"+"
ELSE 1890
1880 C(13),C(12)=0
1890 IF C(0)>=1 AND C(4)>=1
THEN ERG$=ERG$&R$(46)&"+" EL
SE 1910
1900 C(0),C(4)=0
1910 IF C(2)>=1 AND C(6)>=1
THEN ERG$=ERG$&R$(47)&"+" EL
SE 1930
1920 C(2),C(6)=0
1930 IF C(10)>=1 AND C(14)>=
1 THEN ERG$=ERG$&R$(48)&"+"
ELSE 1950
1940 C(10),C(14)=0
1950 IF C(8)>=1 AND C(12)>=1
 THEN ERG$=ERG$&R$(49)&"+" E
LSE 1970
1960 C(8),C(12)=0
1970 IF C(0)>=1 AND C(2)>=1
THEN ERG$=ERG$&R$(50)&"+" EL
SE 1990
1980 C(0),C(2)=0
1990 IF C(1)>=1 AND C(3)>=1
THEN ERG$=ERG$&R$(51)&"+" EL
SE 2010
2000 C(1),C(3)=0
2010 IF C(5)>=1 AND C(7)>=1
THEN ERG$=ERG$&R$(52)&"+" EL
SE 2030
2020 C(5),C(7)=0
2030 IF C(4)>=1 AND C(6)>=1
THEN ERG$=ERG$&R$(53)&"+" EL
SE 2050
2040 C(4),C(6)=0
2050 IF C(2)>=1 AND C(10)>=1
 THEN ERG$=ERG$&R$(54)&"+" E
LSE 2070
2060 C(2),C(10)=0
2070 IF C(3)>=1 AND C(11)>=1
 THEN ERG$=ERG$&R$(55)&"+" E
LSE 2090
2080 C(3),C(11)=0
2090 IF C(7)>=1 AND C(15)>=1
 THEN ERG$=ERG$&R$(56)&"+" E
LSE 2110
2100 C(7),C(15)=0
2110 IF C(6)>=1 AND C(14)>=1
 THEN ERG$=ERG$&R$(57)&"+" E
LSE 2130
2120 C(6),C(14)=0
2130 IF C(10)>=1 AND C(8)>=1
 THEN ERG$=ERG$&R$(58)&"+" E
LSE 2150
2140 C(10),C(8)=0
2150 IF C(11)>=1 AND C(9)>=1
 THEN ERG$=ERG$&R$(59)&"+" E
LSE 2170
2160 C(11),C(9)=0
2170 IF C(15)>=1 AND C(13)>=
1 THEN ERG$=ERG$&R$(60)&"+"
ELSE 2190
```

```
2180 C(15),C(13)=0
2190 IF C(14)>=1 AND C(12)>=
1 THEN ERG$=ERG$&R$(61)&"+"
ELSE 2210
2200 C(14),C(12)=0
2210 IF C(0)>=1 AND C(8)>=1
THEN ERG$=ERG$&R$(62)&"+" EL
SE 2230
2220 C(0),C(8)=0
2230 IF C(1)>=1 AND C(9)>=1
THEN ERG$=ERG$&R$(63)&"+" EL
SE 2250
2240 C(1),C(9)=0
2250 IF C(5)>=1 AND C(13)>=1
 THEN ERG$=ERG$&R$(64)&"+" E
LSE 2270
2260 C(5),C(13)=0
2270 IF C(4)>=1 AND C(12)>=1
 THEN ERG$=ERG$&R$(65)&"+" E
LSE 2300
2280 C(4),C(12)=0
2290 !
2300 !1er BLOCK
2310 FOR I=0 TO 15 :: IF C(I
)=1 THEN ERG$=ERG$&R$(66+I)&
"+"
2320 NEXT I
2330 ERG$=SEG$(ERG$,1,LEN(ER
G$)-1)
2340 !
2350 !ERGEBNISAUSGABE
2360 A$="ERGEBNIS:" :: FOR I
=1 TO LEN(A$):: CALL HCHAR(2
0,16+I,ASC(SEG$(A$,I,1))):: 
NEXT I
2370 ERG$="A="&ERG$ :: IF LE
N(ERG$)>28 THEN 2390
2380 FOR I=1 TO LEN(ERG$)::
CALL HCHAR(22,3+I,ASC(SEG$(E
RG$,I,1))):: NEXT I :: GOTO
2420
2390 TEIL2$=SEG$(ERG$,POS(ER
G$,"+",25)+1,(LEN(ERG$)-POS(
ERG$,"+",25))):: TEIL1$=SEG$
(ERG$,1,POS(ERG$,"+",25))
2400 FOR I=1 TO LEN(TEIL1$):
: CALL HCHAR(22,3+I,ASC(SEG$
(ERG$,I,1))):: NEXT I
2410 FOR I=1 TO LEN(TEIL2$):
: CALL HCHAR(24,3+I,ASC(SEG$
(TEIL2$,I,1))):: NEXT I
2420 A$="NEUE EINGABE ? J/N"
 :: FOR I=1 TO LEN(A$):: CAL
L HCHAR(17,14+I,ASC(SEG$(A$,
I,1))):: NEXT I
2430 CALL KEY(0,K,S):: IF S=
0 THEN 2430
2440 IF K=78 THEN 2450 ELSE
IF K=74 THEN 260 ELSE 2430
2450 CALL CLEAR :: END
```

BÖRSE

Sehr günstig ! Ex-Box mit RS 232, 32K, Disk und Controler 1.099,– Module Datenverw. und Text- und Daeiverwaltung je 40,–. J. Weiß, Franckestr. 14, 3380 Goslar

---

Nur 399,– DM VB für TI-99/4A mit Exbasic + Recorder + Kabel + Lit. + 2 Joyst. + Lerncas. + Softw. Tel. 02850/7654

Suche folgende Module: Pole Position, Popeye, Frogger, Early Learning Fun! Telefon: 0212/ 62443, Blasberg

---

TI99/4A + Exbasic + 6 Mod + Joyst. + Sprachsynth. + Kabel + Rec + Joyadapter + 2 Software Kass. + Bücher. 400 DM! Tel. 06301/4809

Verk. neuw. ext. RSZ32- Interf. f. TI, eingeb. Netzt., Bus durchgef. 2ser. Ports.: DM 220,– + NN/Verp. T.: 069 30 66 71 (nicht v. 7.–25.9.)

TI99/4A + Assembler + TI-Forth + Ex-Basic + E-Box + Disk-Laufwerk + Controler VB 1300 DM 0761/28 13 21

Verk. TI 99/4A + Ex-B. + Rec. Kabel + Recorder + ca. 40 Prg. + 6xTI-Revue VB 600 DM. M. Meincke, Tel. 04105/23 56

TAUSCHE orig. Schachmodul + Publikation „TI 99/4A Intern" + sehr gute ExB.-Software gegen Sprachsynthesizer. Interessenten bitte melden bei Holger Schwenke, Weststr. 33A, 3162 Uetze

---

Suceh Kontakt zu Ti-User. Ist das nichts? Schreibt an: M. Kugelmann, Nassauerstr. 4, 6272 Niedernhausen

---

Verkaufe TI + Box + 32K + RS232 + 2 Laufw. + Contr. + XB + EA + Sprachs. + Multipaln + Buchungsj. + ca. 25 Module + Recorder (alles org.) + Lit + Software auf 100 Disks auch einzeln, Preis VB. Tel. 0821/63 857 (helfe bei Problemen) + Joysticks

---

TI-Module Datenverwaltung + Analyse und Statistik je 55 DM zus. 100 DM. Lutz Klaus, Tel. 0931/95298

Doppel-Slimline-Flop pys VB 02624/38 17

Suche Disk und Controler für Ti99 – Preiswert! Angebot an: Günter Link, Schulstr. 43, 5241 Scheuerfeld

---

TI99/4A + ExtB + Rec + Joyst + 10 Kass = 460 D M Module Schach + Adventure je 50 DM. Buchjournal 100 DM ab 19.00 Uhr 08551 / 1526

Suche Ti-99 User im Raum Lippstadt und Soest zum Prog. und Erfahrungsaustausch in Ti- und Ext. Basic u. Speechsynthesizer (nur Cass.) mögl. Markus Loerwald, Eickelbornstr. 34, 478 Lippstadt 16. Tel. 02945/5711

Verk. TI + Box + 32k + RS 232 + Contr. + Laufw. + 25 Module + Multiplan + 2 Laufwerk. J. Storn, Christleseeweg 34, 8900 Augsburg

---

Verkaufe original 32k intern für Peri-Box – fast neu – für 300,– DM. Tel. 08192/231 oder 089/ 69 16 517

Suche für TI99/4A PAL-Modulator. C. Hasselkuss, Hinter der Mauer 26, 7024 Filderstadt, Tel. 07158/2637, nur Sa-So

---

```
100 !**********************
110 !*     HEXDUMP          *
120 !* Copyright 1985 by   *
130 !*  Bernard Schmidt    *
140 !*                     *
150 !* Benoetigte Geraete :*
160 !*  TI99/4A Konsole    *
170 !* TI Extended Basic   *
180 !*                     *
190 !* Speicherbelegung :  *
200 !*    1633 Bytes       *
210 !*                     *
220 !**********************
230 CALL CLEAR
240 DISPLAY AT(8,3)BEEP:"HEX
DUMP DRUCKERAUSGABE "
250 DISPLAY AT(12,3):"FILENA
ME......:" :: ACCEPT AT(12,1
9):DRUCK$
260 IF DRUCK$="" THEN 250
270 DISPLAY AT(14,3):"ANFANG
ADRESSE :>" :: ACCEPT AT(14,
19)SIZE(4)VALIDATE(DIGIT,"AB
CDEF"):ANF$
280 DISPLAY AT(16,3):"ENDADR
ESSE....:>" :: ACCEPT AT(16,
19)SIZE(4)VALIDATE(DIGIT,"AB
CDEF"):END$
290 OPEN #1:DRUCK$ :: PRINT
#1:"......HEXDUMP" :: PRINT
#1 :: PRINT #1
300 PRINT #1:"......VON..>";
ANF$;"..BIS..>";END$
310 PRINT #1 :: PRINT #1 ::
CLOSE #1
320 CALL HEXDEZ(ANF,ANF$)::
CALL HEXDEZ(ENDE,END$)
330 DIF=ENDE-ANF :: X=INT(DI
F/16):: K=0 :: ADR=ANF
340 FOR I=1 TO X :: A$="" ::
 FOR J=0 TO 15 :: CALL PEEK(
ANF+K+J,WERT)
350 CALL DEZHEX(WERT,WERT$):
: A$=A$&" "&SEG$(WERT$,3,2):
: NEXT J
360 CALL DEZHEX(ADR,ADR$)
370 OPEN #1:DRUCK$ :: PRINT
#1:" >";ADR$;"..";A$ :: CLOS
E #1
380 K=K+16 :: ADR=ADR+16 ::
NEXT I
390 IF DIF-(X*16)>=0 THEN 40
0 ELSE 450
400 Y=DIF-(X*16):: A$="" ::
FOR I=0 TO Y
410 CALL PEEK(ANF+K+I,WERT):
: CALL DEZHEX(WERT,WERT$)
420 A$=A$&" "&SEG$(WERT$,3,2
):: NEXT I
430 CALL DEZHEX(ANF+K,ADR$)
440 OPEN #1:DRUCK$ :: PRINT
#1:" >";ADR$;"..";A$ :: CLOS
E #1
450 CALL CLEAR
460 DISPLAY AT(8,3)BEEP:"NEU
E EINGABE (J/N) ?"
470 CALL KEY(0,K,S):: IF S=0
 THEN 470
480 IF K=74 THEN 230 ELSE IF
 K=78 THEN END ELSE 470
490 SUB HEXDEZ(DEZ,HEX$)
500 K(1)=4096 :: K(2)=256 ::
 K(3)=16 :: K(4)=1
510 ZAHL$="0123456789ABCDEF"
520 DEZ=0 :: HEX$="0000"&HEX
$ :: HEX$=SEG$(HEX$,LEN(HEX$
)-3,4)
530 FOR N=1 TO 4 :: DEZ=(POS
(ZAHL$,SEG$(HEX$,N,1),1)-1)*
K(N)+DEZ
540 NEXT N :: SUBEND
550 SUB DEZHEX(DEZ,HEX$)
560 ZAHL$="0123456789ABCDEF"
 :: HEX$,HEX1$="" :: DEZ1=DE
Z
570 FOR N=1 TO 4 :: R=(DEZ1-
INT(DEZ1/16)*16):: DEZ1=INT(
DEZ1/16)
580 HEX1$=HEX1$&SEG$(ZAHL$,R
+1,1):: NEXT N
590 FOR N=1 TO 4 :: HEX$=HEX
$&SEG$(HEX1$,5-N,1):: NEXT N
 :: SUBEND
```

```
 HEXDUMP

 VON  >0000  BIS  >0036

>0000   83 E0 00 24 02 C0 09 00 83 C0 0A 92 30 AA 04 60
>0010   07 D2 00 78 1E 00 04 60 00 7A 1E 00 04 60 00 78
>0020   04 60 04 B2 02 0D 78 00 02 0E 01 00 02 0F 8C 02
>0030   02 00 00 20 10 13 10
```

```
100 !*********************
110 !* POKELISTENGENERATOR *
120 !* Copyright 1985 by  *
130 !*   Bernard Schmidt   *
140 !*                     *
150 !* benoetigte Geraete: *
160 !*  TI 99/4A Konsole   *
170 !*  TI Extended Basic  *
180 !*   32K-Erweiterung   *
190 !*                     *
200 !*  Speicherbelegung:  *
210 !*     1390 Bytes      *
220 !*                     *
230 !*********************
240 CALL CLEAR
250 DISPLAY AT(8,3)BEEP:"ERS
TELLUNG VON POKELISTEN" :: D
ISPLAY AT(10,8):"MIT DRUCKER
 !"
260 DISPLAY AT(16,3):"FILENA
ME    :" :: ACCEPT AT(16,19)
:DRUCK$ :: IF DRUCK$="" THEN
 260
270 DISPLAY AT(17,3):"PROGRA
MMNAME ?" :: ACCEPT AT(17,19
):NAME$
280 DISPLAY AT(18,3):"STARTA
DRESSE:  >" :: ACCEPT AT(18,
19)SIZE(4)VALIDATE(DIGIT,"AB
CDEF"):ANF$
290 DISPLAY AT(20,3):"ENDADR
ESSE  :  >" :: ACCEPT AT(20,
19)SIZE(4)VALIDATE(DIGIT,"AB
CDEF"):END$
300 OPEN #1:DRUCK$ :: PRINT
#1:"PROGRAMMNAME :";NAME$ ::
 PRINT #1 :: PRINT #1:"POKEL
ISTE VON >";ANF$;" BIS >";EN
D$
310 PRINT #1
320 PRINT #1 :: CLOSE #1
330 CALL HEXDEZ(ANF,ANF$)::
CALL HEXDEZ(ENDE,END$)
340 DIF=ENDE-ANF :: X=INT(DI
F/20):: K=0
350 FOR I=1 TO X :: A$=""
360 FOR J=1 TO 20 :: CALL PE
EK(ANF+K+J-1,WERT):: A$=A$&"
 "&STR$(WERT):: NEXT J
370 OPEN #1:DRUCK$
380 PRINT #1:" DATA  ";A$
390 CLOSE #1
400 K=K+20 :: NEXT I
410 IF DIF-(X*20)>=0 THEN 42
0 ELSE 470
420 Y=DIF-(X*20):: A$="" ::
FOR I=0 TO Y
430 CALL PEEK(ANF+K+I,WERT):
: A$=A$&" "&STR$(WERT):: NEX
T I
440 OPEN #1:DRUCK$
450 PRINT #1:" DATA  ";A$
460 CLOSE #1
470 CALL CLEAR
480 DISPLAY AT(8,3)BEEP:"NEU
E EINGABE (J/N) ?"
490 CALL KEY(0,K,S):: IF S=0
 THEN 490
500 IF K=74 THEN 240 ELSE IF
 K=78 THEN END ELSE 490
510 SUB HEXDEZ(DEZ,HEX$)
520 K(1)=4096 :: K(2)=256 ::
 K(3)=16 :: K(4)=1
530 ZAHL$="0123456789ABCDEF"
540 DEZ=0 :: HEX$="0000"&HEX
$ :: HEX$=SEG$(HEX$,LEN(HEX$
)-3,4)
550 FOR N=1 TO 4 :: DEZ=(POS
(ZAHL$,SEG$(HEX$,N,1),1)-1)*
K(N)+DEZ
560 NEXT N :: SUBEND
```


UNSER
☎ TELEFONSERVICE ☎

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite

ACHTUNG !!!

Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13


```
PROGRAMMNAME :PROBE

POKELISTE VON >0000 BIS >0024

 DATA  -31 224 0 36 131 192 9 0 131 192 10 146 48 170 4 96 2 178 0 0
 DATA  30 0 4 96 0 122 30 0 4 96 0 120 4 96 4 178 2
```

SPASS SPIEL SPANNUNG

IRVEL

Mit Ihrem Raumschiff haben Sie sich auf dem Planeten Irvel verirrt. Um nach dem richtigen Heimweg zu fragen, verlassen Sie Ihr Schiff und erkunden den Planeten zu Fuß. Dabei geraten Sie in das gefürchtete Labyrinth von Irvel.

Zwar finden Sie hier reichlich Schätze vor, mit denen Sie Punkte sammeln können, doch nach 2000 Punkten wird es den Hütern der Schätze zu viel und sie versuchen, den Eindringling zu vernichten.

Die Mauern des Labyrinthes verschieben sich plötzlich und gefährliche Fallgruben entstehen.

Trotzdem kann das Labyrinth erst verlassen werden, wenn mindestens Schätze im Wert von 10 000 Punkten gesammelt wurden. Je höher die Punktzahl steigt, desto gefährlicher wird der Aufenthalt im Labyrinth.

Um Ihnen den Aufenthalt etwas zu erleichtern, bekommen Sie bei 4 500 Punkten einen Bonus und ein zusätzliches Leben.

Nur wenige sind aus diesem Labyrinth wieder heil ans Tageslicht gekommen.

Ob Sie nun auch dazu gehören werden ???

P.S.: Denkt daran: Die Hilfspunkte im Listing nicht mit abtippen.

H.K.

The Castle of 81 Rooms

Dieses Spiel ist im Adventure-Stil aufgebaut und hat eine 3D-Graphik. Die Aufgabe ist ganz einfach: Sie müssen aus dem Labyrinth herausfinden. Das müssen Sie aber lebendig bewerkstelligen. In der Burg mit den 81 Räumen (9x9) gibt es auch böse Geister und einen sogenannten ‚Burgvirus'. Der Kontakt mit einem von beiden endet für Sie tödlich, es sei denn, Sie haben vorher in einem der Räume eine Bibel gegen den Geist oder eine Pille aufgenommen. Nehmen Sie einen Gegenstand aus einem Raum auf, so verschwindet er von dort. Sie können jeweils nur drei Pillen und drei Bibeln tragen. Bei jedem ‚Feindkontakt' wird Ihnen das jeweilige Gegenmittel abgezogen. Truhen und Kelche voller Energie können Sie nehmen, soviel Sie wollen. Sinkt die Energie unter Null oder begegnen Sie einem Geist oder einem Virus ohne das entsprechende Gegenmittel, so sterben Sie.

G.S.

Ecarte

Dieses Programm erlaubt es, gegen den Computer Ecarte zu spielen. Hierbei gelten folgende Spielregeln: Es wird mit einem Skatspiel (32 Karten) gespielt. Die Reihenfolge ist: König, Dame, Bube, As, 10, 9, 8, 7.

Zuerst wird per Zufall bestimmt, welcher der beiden Spieler Geber und welcher Ausspielender ist. Dann werden die Karten gegeben (dies wird unabhängig von den Spielerbezeichnungen immer vom Computer erledigt.). Jeder erhält fünf Karten. Die elfte Karte wird offen auf den Tisch gelegt und bestimmt die Trumpffarbe. Ist es ein König, so erhält der Geber einen Punkt. Auch wenn einer der beiden Spieler den Trumpfkönig hat oder ihn später erhält, so kann er ihn anmelden und erhält einen Punkt dafür. (Taste ‚a').

Nun muß der Ausspielende entscheiden, ob er spielen will (Taste ‚s') oder ob er einige Karten austauschen will (Taste ‚k'). Will er Karten austauschen, so muß der Geber entscheiden, ob er doch spielen soll oder die Karten tauschen kann. dies wird wieder durch dieselben Tasten gesteuert. Bestätigt er den Austausch, so müssen zuerst der Ausspielende und dann der Geber die Zahl und die Nummern der auszutauschenden Karten angeben. Hierbei denkt man sich die Karten von oben nach unten von 1-5 durchnumeriert. Dies ist nur möglich, solange noch Karten auf dem Talon liegen.

Hat einer der Spieler entschieden, daß das Spiel beginnen soll, so spielt der Ausspielende die erste Karte auf. Von diesem Zeitpunkt an ist es nicht mehr möglich, enen Trumpfkönig anzumelden. Für das Bedienen gelten folgende Regeln:

– Man muß, sofern möglich, immer bedienen!

– Jeder Stich muß, wenn möglich, mitgenommen werden!

Eine Karte wird durch Eingabe ihrer Nummer ausgespielt. Nach fünf Stichen ist das Spiel zuende.

Hat derjenige, der zum Spiel aufgefordert hat, alle Stiche gemacht, so bekommt er zwei Punkte; hat er mindestens drei Stiche gemacht, so erhält er einen Punkt und hat er weniger als drei, werden ihm zwei Punkte abgezogen.

Der Computer überwacht, daß der Spieler die Regeln genau einhält und spielt den Stich selbst, wenn man das nicht tut.

Sky-Driver

Sky-Diver ist ein Spiel, das auf dem TI-Grundgerät läuft. Aufgabe des Spielers ist es, Fallschirmspringer aus einem Hubschrauber so abzusetzen, daß diese die Landemarke möglichst in der Mitte treffen.

Die Position der Landemarke sowie der Hindernisse (Bäume und eine große Wolke) werden

durch die RND-Funktion bestimmt.

Anfangs stehen 5 Springer zur Verfügung. Wird die Landemarke in der Mitte getroffen, so erhält der Spieler 3 Punkte; die Anzahl der Springer bleibt unverändert. Wird die Marke am Rand getroffen, so erhält man einen Punkt, aber es wird ein Springer abgezogen. Wird die Marke nicht getroffen, so wird ebenfalls ein Springer abgezogen.

Bei 30 erreichten Punkten bekommt man einen Bonus-Springer. Der Spieler kann zwischen einer leichteren Spielvariante (ohne Windbeeinflussung des Springers) und einer etwas schwierigeren wählen.

Variablen-Liste

I,K,L	Schleifenvariable
M,Y	Merker für Auszuführende bzw. nicht mehr auszuführende Programmteile
A,B	zum Einlesen der Data-Zeilen
SP	Spielart
P	Punkte
MEN	Anzahl der noch vorhandenen Springer
BZ	Merker, ob der Bonus-Springer bereits einmal zugezählt wurde
WM	Wind von links/ rechts bzw. kein Wind
Z,Q,W	Variablen für die GChar-Abfrage
L,U	Bestimmen die Sprungbewegung
PU	Abfrage ob 0,1 oder 3 Punkte erzielt wurden
T	Bestimmt, ob ein Spieler abgezogen wird oder nicht
EOF	

Dirk Junghans

CATCH WITH CAR

Catch with Car ist ein Spielprogramm für den Ti 99/4A in der Grundausstattung. Um das Programm laufen zu lassen, werden folgende Hardware-„Teile" benötigt:
1 Ti99/4A Computer
1 Joystik (Joystik Nr. 2)
1 Fernseher/Monitor
und weiter empfehle ich einen Cassettenrekorder oder eine Diskettenstation zum Abspeichern des Programms.

Sie leben auf einem fremden Planeten, der von riesigen Insekten beherrscht wird. Ihre Aufgabe ist es nun, mit Joystik 2 ein Fahrzeug zu steuern und dabei die Insekten, die immer vereinzelt auftauchen, zu überfahren. Da es auf dem Planeten ausschließlich Sandbogen gibt, reißt Ihr Fahrzeug tiefe Furchen in den Sand. Wenn Sie nun in eine solche Furche hineinfahren, gibt es Punktverlust. Insgesamt dürfen Sie auch nur 15-mal in eine solche Furche fahren.

Es gibt verschiedene Arten von Insekten auf dem Planeten, bei einigen gibt es nur Punkte, wenn sie diese überfahren, bei anderen verschwindet an einigen vereinzelten Stellen die Furche und bei noch anderen gibt es Punkte und es verschwinden Furchenabschnitte. Ihnen stehen drei Fahrzeuge zur Verfügung, diese Zahl vermindert sich, wenn Sie aus dem Spielfeld fahren. Folgende Punkte führen zum Spielende:

Mehr als 15mal in eine Furche gefahren, weniger als Null Punkte. Dreimal aus dem Spielfeld gefahren.

Wenn Sie 1500 Punkte erreicht haben, erhöht sich die Anzahl der Fahrzeuge um eins, 25 Furchenabschnitte werden gelöscht und es ertönt eine Melodie.

Zu Beginn des Spiels und wenn Sie aus dem Feld gefahren sind, müssen Sie den Feuerknopf betätigen, um weiterfahren zu können. Und nun wünsche ich Ihnen viel Spaß mit dem Spiel!!!

Dirk Junghans

Dreiecks- und Kreisberechnungen

Mit diesem Programm stellt Euch der Autor, Jürgen Brühne, eine kleine Hilfe für die Hausaufgaben zur Verfügung. Seid Ihr in Mathe gerade bei dem Thema Geometrie angelangt, kann der Rechner nun Eure Hausaufgaben kontrollieren. Natürlich sollten die Aufgaben auch weiterhin alleine gemacht werden, denn wenn der TI99 Eure Aufgaben alleine macht, bleibt von dem Stoff bei Euch wenig hängen und am Ende des Schuljahres schlägt sich dies in Eurem Notenspiegel nieder. Auf jeden Fall dürft Ihr Euren Rechner aber die Aufgaben auf korrekte Lösungen überprüfen lassen, um am nächsten Tag nicht auf die Nase zu fallen. Da das Programm vollständig in TI-Basic geschrieben wurde, braucht keiner von Euch darauf zu verzichten.

Ein Menue führt durch das Programm und es müssen nur noch die einzelnen Werte der Berechnungen eingegeben werden.

Jürgen Brühne

Dreidimensionale Mühle

Der Sinn dieses Spieles ist es, möglichst viele „Mühlen" zu bauen. 2 Spieler erhalten je 32 Steine, die sie abwechselnd setzen — einer dieser Spieler kann auch der Computer sein — Man versucht nun, durch geschicktes Setzen der Steine jeweils 4 in eine senkrechte, waagerechte oder diagonale Linie zu bringen:

```
0                    0
0                  0
0 0000           0
0                0
```

Dies sind die Mühlen. Wenn alle Steine gesetzt sind, hat der Spieler mit den meisten Mühlen gewonnen. Das Zählen der Mühlen übernimmt der Computer.

Bei diesem Spiel müssen einmal nicht feindliche Objekte abgeschossen werden; es soll vielmehr das logische und räumliche Denken gefördert werden. *Jürgen Brühne*

Copyright Zeichen

Das Programm „Copyright Zeichen" ist in TI-Basic programmiert worden, läuft aber auch unter Extended Basic!

Sicher haben Sie auch schon einmal überlegt, welches Copyrightzeichen Sie für Ihr Programm verwenden wollen, und haben dann krampfhaft irgendein Zeichen definiert. Mit diesem Programm hat das ein Ende! Hier haben Sie die Auswahl zwischen 11 vordefinierten Copyrightzeichen. Wählen Sie in einem Menü Ihr Zeichen und der TI wird Ihnen den 16stelligen Code auf den Bildschirm schreiben! Viel Spaß beim Aussuchen. *Matthias Orf*

```
10 REM*********************
11 REM*                   *
12 REM*      IRVEL        *
13 REM*  Copyright  by    *
14 REM* Manfred Lipowski  *
15 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole  *
19 REM*                   *
23 REM* Speicherbelegung  *
24 REM*   12779 Bytes     *
25 REM*********************
26 REM
120 RANDOMIZE
130 CALL CLEAR
140 CALL SCREEN(7)
150 CALL COLOR(1,16,16)
160 FOR A=1 TO 24
170 PRINT
180 NEXT A
190 REM
       GRAFIK IRVEL
200 CALL CHAR(64,"3C4299A1A1
99423C")
210 CALL CHAR(96,"0018180018
181818")
220 CALL CHAR(97,"1818181818
181818")
230 CALL CHAR(98,"007C7E6666
66667E")
240 CALL CHAR(99,"7C786C6666
666666")
250 CALL CHAR(100,"0066666666
6666666")
260 CALL CHAR(101,"663C3C3C3
C181818")
270 CALL CHAR(102,"007E7E6060
60607C")
280 CALL CHAR(103,"7C6060606
0607E7E")
290 CALL CHAR(104,"0060606060
6060600")
300 CALL CHAR(105,"6060606060
607E7E")
310 REM
       GRAFIK SETZEN
320 CALL COLOR(9,5,16)
330 CALL COLOR(10,5,16)
340 FOR A=2 TO 8
350 CALL COLOR(A,13,16)
360 NEXT A
370 PRINT ".......@ ' b d f
h.......":".........a c e g
i"
380 PRINT : : : : : : : : :
:
390 PRINT "..PRESS ANY KEY T
O BEGIN"
400 DATA 262,330,392,262,330
,392,349,440,523,349,440,523
,262,330,392,262,330,392
410 DATA 196,247,294,196,247
,294,262,330,392,262,330,392
,349,440,523,349,440,523
420 DATA 262,330,392,196,247
,294,262,262,262
430 RESTORE 400
440 FOR A=1 TO 45
450 READ BOT
460 CALL KEY(0,K,S)
470 IF S=0 THEN 480 ELSE 510
480 CALL SOUND(-999,BOT,0,BO
T+2,0,BOT-2,0)
490 NEXT A
500 GOTO 430
510 GOSUB 3250
520 REM   SPIELERKLAERUNG
              ODER
          PROGRAMANLEITUNG
530 FOR A=1 TO 24
540 PRINT
550 NEXT A
560 PRINT "..SPIELERKLAERUNG
 ? (J/N)"
570 PRINT : : : : : : : : :
:
580 GOTO 600
590 CALL SOUND(99,110,0,220,
0)
600 CALL KEY(0,K,S)
610 IF S=0 THEN 600
620 IF K=78 THEN 820
630 IF K=74 THEN 650
640 GOTO 590
650 PRINT ".......@ ' b d f
h":".........a c e g i": :
660 PRINT "IHRE AUFGABE BEST
EHT DARIN": :"DEN ABENTEURER
 SICHER DURCH": :"DAS LABYRI
NTH VON IRVEL ZU": :"FUEHREN
."
670 PRINT :"ER MUSS ALLE 10
SCHAETZE": :"BERUEHREN UM IN
 DIE NAECHSTE": :"RUNDE ZU K
OMMEN. .......>>": : : : :
680 CALL KEY(0,K,S)
690 IF S=0 THEN 680
700 PRINT :"BEI EINEM PUNKTE
STAND VON": :"2000 P WERDEN
DIE TEMPEL-": :"GOETTER ZORN
IG UND FANGEN": :
710 PRINT "AN DAS LABYRINT
ZU VER-": :"SCHIEBEN.MAUERN
WERDEN EIN-": :"GERISSEN UND
 AN ANDERER": :"STELLE WIEDE
R AUFGEBAUT !"
720 PRINT :"LOECHER ERSCHEIN
EN IM BODEN": :"UND VERSCHWI
NDEN NACH EINER": :"WEILE.ER
ST WENN IHR P-KONTO": :
730 PRINT "10000 P ANZEIGT W
```

```
ERDEN SIE": :"AUS DEM LABYRI
NTH ENTLASSEN."
740 CALL KEY(0,K,S)
750 IF S=0 THEN 740
760 PRINT :"DAS SPIEL WIRD M
IT JOYSTICK": :"1 GESTEUERT.
DER ABENTEURER": :"KANN IN J
EDE RICHTUNG ZIEHEN": :
770 PRINT ",DIAGONAL,SENKREC
HT U.S.W.": :"WIRD IHR ABENT
EURER WOHL JE": :"WIEDER DEN
 BLAUEN HIMMEL VON": :"IRVEL
 ZU SEHEN BEKOMMEN ?"
780 PRINT :"WER WEISS...!":
: :"ALPHA LOCK BITTE AUSRAST
EN !"
790 PRINT : : : :
800 CALL KEY(0,K,S)
810 IF S=0 THEN 800
820 CALL CLEAR
830 CALL SCREEN(2)
840 REM
          GRAFIK FUER EINGEN
DLICHES   PROGRAMM
          WAENDE
850 CALL CHAR(33,"3C7EFFFFFF
FF7E3C")
860 CALL CHAR(40,"FF888888FF
2424FF")
870 CALL CHAR(41,"FF212121FF
8989FF")
880 CALL CHAR(42,"FF848484FF
9191FF")
890 REM
          GRAFIK SCHAETZE
900 CALL CHAR(112,"1F2345FDB
58586FC")
910 CALL CHAR(113,"224149556
3777F3E")
920 CALL CHAR(114,"101038101
0101010")
930 CALL CHAR(115,"7F30FC783
0000000")
940 CALL CHAR(116,"0000FE040
E1F0E04")
950 CALL CHAR(117,"183C5A7E7
E243C18")
960 CALL CHAR(118,"1C222A5D4
92A221C")
970 CALL CHAR(119,"000040A0B
FA54500")
980 REM   ABENTEURER
990 CALL CHAR(120,"181866BD9
9242442")
1000 CALL CHAR(121,"181A6FBD
9A242412")
1010 CALL CHAR(122,"1858F6BD
59242448")
1020 REM GRAFIK ABENTEU.ANZE
IGE U. HINTERG.
1030 CALL CHAR(140,"00000000
00000000")
1040 CALL CHAR(141,"181866BD
99242442")
1050 REM   SPIELFELD AUFBAU
1060 CALL COLOR(2,13,15)
1070 CALL COLOR(1,2,16)
1080 CALL COLOR(14,15,1)
1090 CALL COLOR(3,8,1)
1100 CALL COLOR(4,8,1)
1110 CALL COLOR(5,11,1)
1120 CALL COLOR(6,11,1)
1130 CALL COLOR(7,11,1)
1140 CALL COLOR(8,11,1)
1150 CALL COLOR(11,5,16)
1160 CALL COLOR(12,2,16)
1170 CALL HCHAR(1,1,140,96)
1180 DATA "(((((((((((((((((
((((((((((((((","() *) (...
.(....*)....(...( *) ("
1190 DATA "()....(....(....*
)....(...(....(","(((() ( *(
(((() *) *(((() ( *(((","().
.............*).............
("
1200 DATA "((((((((() *((((((
((((((() *(((((","()........
......*).............("
1210 DATA "((() *((((()..( *
) (..*(((() *((","()........
.( () *) *) (........(","(((
(((((() (..(....(..( *((((((
("
1220 DATA "()........*(((((
((((().......(","() *((((((
((((....*(((((((((() ("
1230 DATA "()........(..(..
..(..(........(","((() *((((
() *() *(() *(((() *(("
1240 DATA "()..............
..............(","((((((((()
*(((((((((((() *((((((("
1250 DATA "()..............*
).............(","(((() ( *(
(((() *) *(((() ( *(((("
1260 DATA "()....(...(.....*
)....(...(....(","() *) (...
(.....*)....(...( *) ("
1270 DATA "(((((((((((((((((
((((((((((((((("
1280 RESTORE 1180
1290 FOR A=4 TO 24
1300 READ B$
1310 FOR B=1 TO LEN(B$)
1320 CALL HCHAR(A,B,ASC(SEG$
(B$,B,1)))
1330 NEXT B
1340 NEXT A
1350 A=1
1360 B=3
1370 A$="SCORE:"
1380 GOSUB 2890
```

```
1390 A=1
1400 B=19
1410 A$="MEN:"
1420 GOSUB 2890
1430 MEN=3
1440 GOSUB 2940
1450 SCORE=50
1460 GOSUB 3040
1470 GOSUB 2390
1480 REM  ABENTEURER ZUFAELL
IG        SETZEN !
1490 XX=INT(RND*19)+4
1500 YY=INT(RND*27)+2
1510 CALL GCHAR(XX,YY,PP)
1520 IF PP<>32 THEN 1490
1530 CALL HCHAR(XX,YY,120)
1540 GOSUB 3250
1550 PAT=120
1560 REM  HAUPTTEIL DES
        PROGRAMMS
1570 CALL HCHAR(XX,YY,PAT)
1580 CALL SOUND(-999,-4,9)
1590 IF SCORE<=2000 THEN 162
0 ELSE 1600
1600 IF RND<.4 THEN 1610 ELS
E 1620
1610 GOSUB 2520
1620 CALL JOYST(1,X,Y)
1630 IF (X=0)*(Y=0)THEN 1570
1640 IF X=-4 THEN 1690
1650 IF X=4 THEN 1720
1660 IF Y=-4 THEN 1750
1670 IF Y=4 THEN 1780
1680 GOTO 1620
1690 PAT=121
1700 SOU=-5
1710 GOTO 1800
1720 PAT=122
1730 SOU=-5
1740 GOTO 1800
1750 PAT=120
1760 SOU=-6
1770 GOTO 1800
1780 PAT=120
1790 SOU=-7
1800 CALL HCHAR(XX,YY,32)
1810 SCORE=SCORE-1
1820 IF SCORE<0 THEN 3310
1830 XX=XX+(Y/-4)
1840 YY=YY+(X/4)
1850 CALL GCHAR(XX,YY,PP)
1860 IF PP<>32 THEN 1900
1870 CALL HCHAR(XX,YY,PAT)
1880 CALL SOUND(-1,SOU,9)
1890 GOTO 1620
1900 REM    ABFRAGE OB WAND
 ODER SCHATZ BERUEHRT
1910 IF (PP=40)+(PP=41)+(PP=
42)+(PP=33)THEN 1920 ELSE 21
10
1920 FOR A=3 TO 30 STEP 3
1930 CALL SOUND(-99,-5,A)
1940 NEXT A
1950 CALL HCHAR(XX,YY,40)
1960 MEN=MEN-1
1970 A=3
1980 B=3
1990 A$="DORT"&CHR$(140)&"WA
R"&CHR$(140)&"EINE"&CHR$(140
)&"MAUER"
2000 IF PP=33 THEN 2010 ELSE
 2020
2010 A$="INS"&CHR$(140)&"LOC
H"&CHR$(140)&"GEFALLEN"
2020 GOSUB 2890
2030 GOSUB 3250
2040 GOSUB 2940
2050 SCORE=SCORE-100
2060 GOSUB 3040
2070 FOR A=1 TO 400
2080 NEXT A
2090 CALL HCHAR(3,1,140,32)
2100 GOTO 1490
2110 CALL HCHAR(XX,YY,PAT)
2120 TONO=INT(RND*220)+220
2130 FOR A=0 TO 30 STEP 5
2140 CALL COLOR(12,7,16)
2150 CALL SOUND(-300,TONO,A,
TONO+2,A,TONO/2,A)
2160 CALL COLOR(12,2,16)
2170 NEXT A
2180 IF PP=117 THEN 2190 ELS
E 2210
2190 SCORE=SCORE+150
2200 GOTO 2320
2210 IF PP=113 THEN 2220 ELS
E 2240
2220 SCORE=SCORE+125
2230 GOTO 2320
2240 IF PP=118 THEN 2250 ELS
E 2270
2250 SCORE=SCORE+200
2260 GOTO 2320
2270 IF PP=112 THEN 2280 ELS
E 2310
2280 SCOR1=INT(RND*150)+25
2290 SCORE=SCORE+SCOR1
2300 GOTO 2320
2310 SCORE=SCORE+100
2320 GOSUB 3040
2330 SCOR1=0
2340 SCHATZ=SCHATZ+1
2350 IF SCHATZ=10 THEN 2360
ELSE 1620
2360 GOSUB 2390
2370 GOTO 1570
2380 REM   SCHATZ SETZEN
2390 FOR AP=1 TO 10
2400 QWE=INT(RND*220)+220
2410 QQ=INT(RND*19)+4
```

```
2420 WW=INT(RND*29)+2
2430 CALL GCHAR(QQ,WW,EE)
2440 IF EE<>32 THEN 2410
2450 RR=INT(RND*8+1)+111
2460 CALL HCHAR(QQ,WW,RR)
2470 CALL SOUND(-99,QWE,AP*2
,QWE/2,AP*2)
2480 NEXT AP
2490 SCHATZ=0
2500 RETURN
2510 REM     WAND ENTFERNEN
             LOCH SETZEN
             LOCH ENTFERNEN
             WAND SETZEN
2520 REM
2530 ZX=INT(RND*18)+5
2540 XZ=INT(RND*27)+3
2550 CALL GCHAR(ZX,XZ,ZZ)
2560 IF (ZZ<>40)*(ZZ<>41)*(Z
Z<>42)THEN 2530
2570 CALL SOUND(-99,-7,0)
2580 CALL HCHAR(ZX,XZ,32)
2590 CALL SOUND(-99,-5,3)
2600 ON INT(RND*3+1)GOTO 261
0,2680,2790
2610 ZX=INT(RND*18)+5
2620 XZ=INT(RND*27)+5
2630 CALL GCHAR(ZX,XZ,ZZ)
2640 IF ZZ<>32 THEN 2610
2650 CALL SOUND(-99,-5,0)
2660 CALL HCHAR(ZX,XZ,33)
2670 CALL SOUND(-99,-6,3)
2680 IF RND<.4 THEN 2690 ELS
E 2790
2690 ZX=INT(RND*18)+5
2700 XZ=INT(RND*27)+5
2710 XTA=XTA+1
2720 IF XTA=5 THEN 2780
2730 CALL GCHAR(ZX,XZ,ZZ)
2740 IF ZZ<>33 THEN 2690
2750 CALL SOUND(-99,-6,0)
2760 CALL HCHAR(ZX,XZ,40)
2770 CALL SOUND(-99,-7,5)
2780 XTA=0
2790 IF RND<.6 THEN 2800 ELS
E 2870
2800 ZX=INT(RND*18)+5
2810 XZ=INT(RND*27)+5
2820 CALL GCHAR(ZX,XZ,ZZ)
2830 IF ZZ<>32 THEN 2800
2840 CALL SOUND(-99,-1,0)
2850 CALL HCHAR(ZX,XZ,40)
2860 CALL SOUND(-99,-3,0)
2870 RETURN
2880 REM WOERTER PRINTEN
2890 FOR C=1 TO LEN(A$)
2900 CALL HCHAR(A,B+C,ASC(SE
G$(A$,C,1)))
2910 CALL SOUND(1,-1,0)
2920 NEXT C
2930 RETURN
2940 IF MEN<=0 THEN 2990 ELS
E 2950
2950 CALL HCHAR(1,24,140,4)
2960 CALL HCHAR(1,24,141,MEN
)
2970 CALL SOUND(-1,-2,6)
2980 RETURN
2990 CALL HCHAR(1,24,48)
3000 GOSUB 3040
3010 GOSUB 3250
3020 GOTO 3310
3030 REM SCORE PRINTEN
3040 CALL HCHAR(1,10,140,6)
3050 FOR AFF=1 TO LEN(STR$(S
CORE))
3060 CALL HCHAR(1,9+AFF,ASC(
SEG$(STR$(SCORE),AFF,1)))
3070 CALL SOUND(-1,-3,5)
3080 NEXT AFF
3090 IF SCORE<=0 THEN 3310
3100 IF (SCORE>4500)*(SCORE<
5500)THEN 3110 ELSE 3220
3110 SCORE=SCORE+2500
3120 A=3
3130 B=3
3140 A$="BONUS"&CHR$(140)&"B
ONUS"&CHR$(140)&"BONUS"&CHR$
(140)&"BONUS"
3150 GOSUB 2890
3160 MEN=MEN+1
3170 GOSUB 2940
3180 GOSUB 3250
3190 GOSUB 3250
3200 GOSUB 3250
3210 CALL HCHAR(3,1,140,32)
3220 IF SCORE>=10000 THEN 37
10
3230 RETURN
3240 REM   ZUFALLSERZEUGUNG
                TON
3250 TON=INT(RND*280)+220
3260 FOR ATT=0 TO 30 STEP 2.
5
3270 CALL SOUND(-99,TON,ATT,
TON+2,ATT,TON/2,ATT)
3280 NEXT ATT
3290 RETURN
3300 REM   SPIELENDE
3310 FOR A=2 TO 28
3320 CALL COLOR(1,2,7)
3330 CALL COLOR(11,2,7)
3340 CALL SOUND(-99,400,A-2,
400+A*2,A,400+A*2,A)
3350 CALL COLOR(1,2,16)
3360 CALL COLOR(11,5,16)
3370 CALL SOUND(-99,400,A-2,
400-A*2,A,400-A*2,A)
3380 NEXT A
3390 FOR A=0 TO 30 STEP 2.5
3400 CALL SCREEN(7)
```

```
3410 CALL SOUND(-99,-5,A)
3420 CALL SCREEN(16)
3430 NEXT A
3440 CALL SCREEN(2)
3450 FOR A=32 TO 1 STEP -1
3460 CALL VCHAR(4,A,140,21)
3470 NEXT A
3480 CALL HCHAR(3,1,140,32)
3490 SCORE=SCORE+150
3500 GOSUB 3040
3510 A=8
3520 B=4
3530 A$="LEIDER"&CHR$(140)&"
VERLOREN"
3540 GOSUB 2890
3550 A=10
3560 B=4
3570 A$="NOCHMAL"&CHR$(140)&
"?"&CHR$(140)&"J"&CHR$(140)&
"N"
3580 GOSUB 2890
3590 RESTORE 400
3600 FOR A=1 TO 45
3610 READ BOT
3620 CALL SOUND(-999,BOT,0,B
OT+2,0,BOT+4,0)
3630 CALL KEY(0,K,S)
3640 IF S=0 THEN 3650 ELSE 3
670
3650 NEXT A
3660 GOTO 3590
3670 IF (K=74)+(K=106)THEN 1
050
3680 GOSUB 3250
3690 CALL CLEAR
3700 END
3710 REM          SPIEL
                GESCHAFFT
3720 FOR A=1 TO 3
3730 FOR B=560 TO 760 STEP 2
0
3740 CALL SOUND(-99,560,A,B,
A,B+2,A)
3750 NEXT B
3760 NEXT A
3770 CALL CLEAR
3780 CALL SCREEN(2)
3790 CALL CHAR(136,"7FDB6E1F
0300000000")
3800 CALL CHAR(138,"C060C0E0
B0D8340E")
3810 CALL CHAR(139,"0300000000
0000000000")
3820 CALL CHAR(128,"0000003C
66FBFEBF")
3830 CALL CHAR(129,"FF6E3C00
0000000000")
3840 CALL CHAR(130,"0000003C
6EFFFF7D")
3850 CALL CHAR(131,"F37E3C00
0000000000")
3860 CALL CHAR(104,"0000030 3
0F0F3CFC")
3870 CALL CHAR(105,"FFFFC3C3
0000000000")
3880 CALL CHAR(106,"0000C0C0
F0F03C3F")
3890 CALL COLOR(14,16,5)
3900 CALL COLOR(13,11,5)
3910 CALL COLOR(10,15,5)
3920 CALL HCHAR(1,1,140,224)
3930 CALL HCHAR(2,4,136)
3940 CALL HCHAR(2,5,138)
3950 CALL HCHAR(3,5,139)
3960 CALL HCHAR(3,20,136)
3970 CALL HCHAR(3,21,138)
3980 CALL HCHAR(4,21,139)
3990 CALL HCHAR(2,28,128)
4000 CALL HCHAR(2,29,130)
4010 CALL HCHAR(3,28,129)
4020 CALL HCHAR(3,29,131)
4030 FOR A=8 TO 24
4040 CALL SOUND(-99,-7,A)
4050 CALL HCHAR(A,1,40,32)
4060 CALL HCHAR(A,32-A,32,2)
4070 NEXT A
4080 FOR A=1 TO 200
4090 NEXT A
4100 ZZ=0
4110 FOR A=24 TO 7 STEP -1
4120 CALL HCHAR(A,9+ZZ,122)
4130 CALL SOUND(2,-7,4)
4140 CALL HCHAR(A,9+ZZ,32)
4150 CALL SOUND(2,-5,1)
4160 ZZ=ZZ+1
4170 NEXT A
4180 CALL COLOR(12,2,5)
4190 CALL HCHAR(7,26,120)
4200 CALL SOUND(2,-6,0)
4210 FOR A=1 TO 99
4220 NEXT A
4230 CALL HCHAR(7,7,104)
4240 CALL HCHAR(7,8,105)
4250 CALL HCHAR(7,9,106)
4260 FOR A=0 TO 30 STEP 5
4270 CALL SOUND(-99,-5,A)
4280 NEXT A
4290 FOR A=26 TO 10 STEP -1
4300 CALL SOUND(3,-7,3)
4310 CALL HCHAR(7,A,121)
4320 CALL SOUND(3,-6,8)
4330 CALL HCHAR(7,A,140)
4340 NEXT A
4350 CALL HCHAR(6,8,120)
4360 CALL SOUND(9,-5,0)
4370 FOR A=8 TO 24
4380 CALL SOUND(-99,-5,A)
4390 CALL HCHAR(A,1,40,32)
4400 CALL SOUND(-99,-6,A)
4410 NEXT A
4420 CALL COLOR(4,2,13)
```

```
4430 CALL COLOR(5,2,13)
4440 CALL COLOR(6,2,13)
4450 CALL COLOR(7,2,13)
4460 CALL COLOR(8,2,13)
4470 A=12
4480 B=4
4490 A$="G(E(S(C(H(A(F(F(T"
4500 GOSUB 2890
4510 A=14
4520 B=4
4530 A$="NOCH(EIN(SPIEL(?(J(
N"
4540 GOSUB 2890
4550 DATA 220,294,294,330,37
0,294,370,330,220,294,294,33
0,370,292,277,277
4560 DATA 220,294,294,330,37
0,392,370,330,294,277,220,24
7,277,294,294
4570 RESTORE 4550
4580 FOR A=1 TO 31
4590 CALL COLOR(10,16,5)
4600 READ B
4610 CALL KEY(0,K,S)
4620 IF S=0 THEN 4630 ELSE 4
670
4630 CALL COLOR(10,15,5)
4640 CALL SOUND(-550,B,0,B+2
,0,B+2,0)
4650 NEXT A
4660 GOTO 4570
4670 IF (K=74)+(K=106)THEN 1
00
4680 CALL CLEAR
4690 END
```

BÖRSE

TI-Peripherie zu kaufen gesucht, auch Z80 bzw. 80-Zeich.-Karte. Johann Schmitz, Postfach 900771, 5 Köln 90

Verk.: 32k-Card 300, P-Code-Card + Fill / ED / COM ass. / Lnk / Uti 700, Disk-Cont + Dos 80 + Manager + 2 Kabel 350,– 06441/74830

Suche preiswert TI-Modul Schach, Minimem, 32K-Erw. seitlich ansteckbar. A. Mader, Tel. 069/ 78 37 25, ab 18 Uhr

---

Kaufe TI 99/4A-Konsole oder Konsole mit Zubehör. H. Ludwig, An Den Tannen 1, 6390 Usingen 1 (06081/2897)

Suche Tauschpartner für Software in TI + Ex Liste an: Andreas Hoyer, Tulpenweg 7, 6052 Mühlheim

---

Verk. Software in TI+Ex ab 1 DM. Suche TI-Special II und Farb-Monitor 06108/69766

Verk. TI 99/4A+XBasic + P-Box + Discontrl. + Laufwerk + 32K + E/A-Mod + Spech-Synth. + Grafik-Tableau + Cas.-Kabel + Deutsches E/A Buch + 10 Disk. + Joyst. + 50 Prg. + Literatur VB 2200 DM. Andreas Hoyer, Tulpenweg 7, 6052 Mühlheim, Tel. 06108/69766

```
10 REM********************
11 REM*                  *
12 REM* COPYRIGHT-ZEICHEN*
13 REM*  Copyright by    *
14 REM*  Matthias Orf    *
15 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                  *
23 REM* Speicherbelegung *
24 REM*    3618 Bytes    *
25 REM********************
26 REM
270 CALL CLEAR
280 CALL CHAR(48,"0038444C54
644438")
290 CALL CHAR(79,"0038444444
444438")
300 CALL CHAR(97,"3C4299A1A1
99423C")
310 CALL CHAR(98,"3C4299A199
423C00")
320 CALL CHAR(99,"FF81B9A1A1
B981FF")
330 CALL CHAR(100,"FF8199A1A
19981FF")
340 CALL CHAR(101,"3C4299919
199423C")
350 CALL CHAR(102,"3C4218202
018423C")
360 CALL CHAR(103,"0042 99A1A
1994200")
370 CALL CHAR(104,"001C202 02
0201C00")
380 CALL CHAR(105,"003C202 02
0203C00")
390 CALL CHAR(106,"E781B9A1A
1B981E7")
400 CALL CHAR(107,"E78199A1A
1A199E7")
410 FOR I=1 TO 24
420 PRINT
430 NEXT I
440 FOR E=1 TO 14
450 CALL COLOR(E,16,11)
460 NEXT E
470 CALL SCREEN(16)
480 PRINT ".....C O P Y R I
G H T": :".......Z E I C H E
 N": : : : : : : : : : : : :
 : : : : : :
490 X=12
500 Y=9
510 T$="MARO-SOFT"
520 GOSUB 2090
530 X=14
540 Y=9
550 T$="BAUNATAL"
560 GOSUB 2090
570 CALL COLOR(9,5,11)
580 CALL COLOR(10,5,11)
```

```
590 FOR I=107 TO 97 STEP -1
600 CALL HCHAR(12,8,I)
610 FOR A=1 TO 130
620 NEXT A
630 NEXT I
640 FOR I=1 TO 2
650 FOR W=200 TO 2000 STEP 3
0
660 CALL SOUND(-50,W,0)
670 NEXT W
680 FOR W=2000 TO 200 STEP -
20
690 CALL SOUND(-50,W,0)
700 NEXT W
710 NEXT I
720 FOR W=0 TO 30 STEP 1
730 CALL SOUND(-200,200,W)
740 NEXT W
750 X=18
760 Y=3
770 T$="SIE HABEN DIE QUAL D
ER"
780 GOSUB 2090
790 X=20
800 Y=2
810 T$="WAHL BEI DEN COPY-ZE
ICHEN"
820 GOSUB 2090
830 FOR I=1 TO 600
840 NEXT I
850 FOR E=1 TO 7
860 FOR W=2 TO 16
870 CALL SCREEN(W)
880 NEXT W
890 NEXT E
900 CALL CLEAR
910 FOR E=1 TO 14
920 CALL SCREEN(11)
930 CALL COLOR(E,7,11)
940 NEXT E
950 CALL COLOR(9,2,11)
960 CALL COLOR(10,2,11)
970 PRINT "...a b c d e f g
h i j  k": : :"...1 2 3 4 5
6 7 8 9 10 11": : : :
980 PRINT "WAEHLEN SIE JETZT
 BITTE IHR": :"...COPYRIGHT-
ZEICHEN UND": :" IHR TI-99/4
A HOMECOMPUTER": :
990 PRINT "SCHREIBT IHNEN DE
N CODE ZUR": :"VERWENDUNG IN
 IHREM EIGENEN": :".......P
ROGRAMM !": : :
1000 INPUT "IHRE WAHL?..(1-1
1)..":W
1010 IF W=1 THEN 1150
1020 IF W=2 THEN 1210
1030 IF W=3 THEN 1270
1040 IF W=4 THEN 1330
1050 IF W=5 THEN 1390
1060 IF W=6 THEN 1450
1070 IF W=7 THEN 1510
1080 IF W=8 THEN 1570
1090 IF W=9 THEN 1630
1100 IF W=10 THEN 1690
1110 IF W=11 THEN 1750
1120 CALL CLEAR
1130 PRINT "SIE SOLLEN DOCH
EINE ZAHL": :"ZWISCHEN ""1""
 UND ""11""": :"WAEHLEN !":
: : :
1140 GOTO 1000
1150 CALL CLEAR
1160 X=10
1170 Y=2
1180 T$="a = ""3C4299A1A1994
23C"""
1190 GOSUB 2090
1200 GOTO 1810
1210 CALL CLEAR
1220 X=10
1230 Y=2
1240 T$="b = ""3C4299A199423
C"""
1250 GOSUB 2090
1260 GOTO 1810
1270 CALL CLEAR
1280 X=10
1290 Y=2
1300 T$="c = ""FF81B9A1A1B98
1FF"""
1310 GOSUB 2090
1320 GOTO 1810
1330 CALL CLEAR
1340 X=10
1350 Y=2
1360 T$="d = ""FF8199A1A1998
1FF"""
1370 GOSUB 2090
1380 GOTO 1810
1390 CALL CLEAR
1400 X=10
1410 Y=2
1420 T$="e = ""3C42999191994
23C"""
1430 GOSUB 2090
1440 GOTO 1810
1450 CALL CLEAR
1460 X=10
1470 Y=2
1480 T$="f = ""3C42182020184
23C"""
1490 GOSUB 2090
1500 GOTO 1810
1510 CALL CLEAR
1520 X=10
1530 Y=2
1540 T$="g = ""004299A1A1994
2"""
1550 GOSUB 2090
1560 GOTO 1810
1570 CALL CLEAR
```

```
1580 X=10
1590 Y=2
1600 T$="h = ""001C2020202
01C"""
1610 GOSUB 2090
1620 GOTO 1810
1630 CALL CLEAR
1640 X=10
1650 Y=2
1660 T$="i = ""003C2020202
03C"""
1670 GOSUB 2090
1680 GOTO 1810
1690 CALL CLEAR
1700 X=10
1710 Y=2
1720 T$="j = ""E781B9A1A1B
981E7"""
1730 GOSUB 2090
1740 GOTO 1810
1750 CALL CLEAR
1760 X=10
1770 Y=2
1780 T$="k = ""E78199A1A1A
199E7"""
1790 GOSUB 2090
1800 GOTO 1810
1810 X=18
1820 Y=2
1830 T$="WENN FERTIG,"
1840 GOSUB 2090
1850 X=20
1860 Y=2
1870 T$="TASTE DRUECKEN"
1880 GOSUB 2090
1890 CALL KEY(0,K,S)
1900 IF S=0 THEN 1890
1910 CALL HCHAR(16,1,32,320)
1920 X=18
1930 Y=2
1940 T$="MOECHTEN SIE NOCH"
1950 GOSUB 2090
1960 X=20
1970 Y=2
1980 T$="MEHR CODE'S SEHEN ?
"
1990 GOSUB 2090
2000 INPUT " ":N$
2010 IF N$="J" THEN 900
2020 IF N$="N" THEN 2030
2030 CALL CLEAR
2040 PRINT ".....- AUF WIEDE
RSEHEN -": : : : : : : : : :
 : :
2050 FOR I=1 TO 500
2060 NEXT I
2070 PRINT : : : : : : : : :
 : : : : :
2080 END
2090 FOR W=1 TO LEN(T$)
2100 CALL HCHAR(X,Y+2,ASC(SE
G$(T$,W,1)))
2110 Y=Y+1
2120 NEXT W
2130 RETURN
```

Die nächste TI REVUE erscheint am 28. Oktober

BÖRSE

Verkaufe TI99/4A + 32 k + Centr. + Recorder + Kabel + Ext.-Basic + Literat. + Listings + Programm-Cass. + Mini-roboter mit Steuerungsprogramm + 2 Joysticks + Modul „Car Wars“
Tel.: 0781 / 75138

HILFE! Suche ganzes TI-99/4A Gehäuse für ca. 15 bis 20 DM. Angebote an Michael v.d. Mühlen, Tel. 02103/44144

Verkaufe Original TI ext. Basic Modul und Sanyo Monitor mit Kabel (grün/28 cm) im Raum Stuttgart für DM 150,– Bzw. 300,–.
Tel. 0711/748318 nach 18 Uhr.

Verkaufe Attack-Modul für 20 DM !!!
Carsten Simms, Oldauer Heuweg 66,
3107 Hambühren 2, Tel. 05084/3578

Verkaufe Drucker Star DP515 DIN A3, Serielle + Paralelle Schnittstelle. Neupreis 2000
Verk.Preis: 600,– DM
Wie funktioniert Spracheing. MBX ohne Module, wer kann mir helfen? (Gegenleistung)
Tel.: 07175/5031 (9–16 Uhr)

Verk. o. tausche Lern- und Spielmodule (Othello u.a.) u. Apesoft Grafikprog. (mit 40 Grafikbefehlen!) Billig! Tel. 0561/516790

Software-Verkauf für 99/4A: Module, X-Bas., Kass. und Bücher. Tel. 04102-64889

Suche gute Assembler Prg. gegen Tausch. Suche Kontakte mit Akkustikkoppler-User.
Tel. 07175/5031

Tausche TI99/4A-Module Alpiner, Carwars, Othello, TI-Invaders gegen Pal-Modulator f. UHF-Kanal 36 – Tel. 05205/7405

Verkaufe: TI99/4A + EXT.-Basic + Handbücher + Joysticks + Kass. Rec.-Kabel (1) + Literatur + Software, alles original TI! UB 450,– DM
Michael Habermann, Tel.: 06842/514 83 nachmittags.

Tausche od. verkaufe: Cassetten HiFi-Deck NAKAMICHI DRAGON (Referenz bei Stereoplay u. Stereo) m. folgd. Ausstattg. = Revers, Einmesscomputer, autom. Azimuth-Regelung usw. Neupr. DM 4150,– im Tausch gegen kpl. ausgest. TI-99/4A Computer-Anlage od. für DM 3600,–
R. Göhring, Alvesloher Str. 21, 2358 Kaltenkirchen
04191/7305 ab 15.10.85

```
10 REM*********************
11 REM*CASTLE of 81 DOOMS*
12 REM*                  *
13 REM*  Copyright by    *
14 REM* Gerhard Stegmann *
15 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                  *
23 REM* Speicherbelegung *
24 REM*    9070 Bytes    *
25 REM*********************
26 REM
40 CALL CLEAR
50 CALL SCREEN(12)
60 PRINT "*********************
*********"
70 PRINT "* THE CASTLE OF 81
 ROOMS *..*(C)BY G.STEGMANN
&......*..*GRAPHIK S.BURKERT
 1/84..*"
80 PRINT "*********************
*********"
90 PRINT : : : : : : : : : :
 : : : : :
100 PRINT "SPIELANLEITUNG?(J
/N)"
110 CALL KEY(0,K,S)
120 IF S=0 THEN 110
130 IF K<>74 THEN 540
140 CALL CLEAR
150 PRINT "SIE BEFINDEN SICH
 IN EINEM":"SCHLOSS MIT 81 R
AEUMEN": :"DIESE RAEUME BEIN
HALTEN"
160 PRINT "VERSCHIEDENE GEGE
NSTAENDE": :"ES KOENNEN JEWE
ILS 2":"GEGENSTAENDE VON EIN
ER ART":"MIT DER TASTE'N'AUF
-"
170 PRINT "GENOMMEN WERDEN":
 :"EIN TIEFER TON NACH DEM "
:"'AKTIONSPIEPS' SIGNALISIER
T"
180 PRINT "DASS NICHTS MEHR
AUFGENOMMEN":"WERDEN KANN."
190 PRINT : :"BITTE EINE TAS
TE DRUECKEN"
200 CALL KEY(0,K,S)
210 IF S=0 THEN 200
220 CALL CLEAR
230 PRINT "DIE BIBEL SCHUETZ
T SIE":"VOR GEISTERN,DIE IHN
EN":"MANCHMAL BEGEGNEN"
240 PRINT "DIE PILLEN SCHUET
ZEN SIE VOR":"DEM'BURGVIRUS'
DER SIE ":"VON ZEIT ZU ZEIT
INFIZIERT"
250 PRINT "KELCHE UND TRUHEN
 KOENNEN ":"SIE SICH NEHMEN,
SOVIEL SIE ":"WOLLEN"
260 PRINT : :"BITTE EINE TAS
TE DRUECKEN"
270 CALL KEY(0,K,S)
280 IF S=0 THEN 270
290 CALL CLEAR
300 PRINT "** DIE PUNKTEVERT
EILUNG **": :
310 PRINT "PUNKTE.I..GEGENST
AND"
320 PRINT "----------------
---"
330 PRINT ".......I"
340 PRINT "..60...I..TRUHE"
350 PRINT "...2...I..KELCH"
360 PRINT "...2...I..PILLE"
370 PRINT "...2...I..BIBEL"
380 PRINT ".......I"
390 PRINT : :"JEDER SCHRITT
ODER JEDE TAT": :"KOSTET SIE
 2 ENERGIE-":"EINHEITEN": :"
DER KELCH BRINGT 10"
400 PRINT "ENERGIEEINHEITEN
ZURUECK": :"BITTE EINE TASTE
 DRUECKEN"
410 CALL KEY(0,K,S)
420 IF S=0 THEN 410
430 CALL CLEAR
440 PRINT "....** DIE STEUER
UNG **": :
450 PRINT ".........V(VORWAER
TS)"
460 PRINT "L(LINKS).........
...R(RECHTS)"
470 PRINT ".........Z(ZURUECK
)"
480 PRINT : :"....N(GEGENSTA
ND AUFNEHMEN)": :"DER 'HINWE
IS' ERINNERT":"SIE DARAN,DAS
S DER AUSGANG"
490 PRINT "IN EINEM DER ANGR
ENZENDEN":"RAEUME IST.SIE MU
ESSEN ":"IHN DANN NUR NOCH F
INDEN"
500 PRINT "ALLES KLAR?(J/N)"
510 CALL KEY(0,K,S)
520 IF S=0 THEN 510
530 IF K<>74 THEN 40
540 CALL CLEAR
550 DIM RAU(9,9)
560 ENE=99
570 PU=0
580 REM ** AUFBAU DER RAEUME
 **
590 FOR RAUM=1 TO 9
600 LET ZAHL(RAUM)=0
610 FOR RAUM2=1 TO 9
620 RANDOMIZE
630 LET RAU(RAUM,RAUM2)=INT(
9*RND+1)
640 IF RAU(RAUM,RAUM2)=2 THE
N 620
```

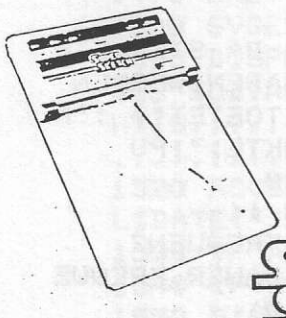
```
650 LET WE(RAUM,RAUM2)=RAU(R
AUM,RAUM2)
660 NEXT RAUM2
670 NEXT RAUM
680 RANDOMIZE
690 HH=INT(9*RND+1)
700 RANDOMIZE
710 BH=INT(9*RND+1)
720 RAU(HH,BH)=2
730 GOSUB 3080
740 GOSUB 2710
750 CALL CLEAR
760 CALL SCREEN(2)
770 FOR C=2 TO 8
780 CALL COLOR(C,16,2)
790 NEXT C
800 FOR Y=1 TO 1500
810 NEXT Y
820 RANDOMIZE
830 HO=INT(9*RND+1)
840 RANDOMIZE
850 BR=INT(9*RND+1)
860 GOSUB 890
870 GOSUB 960
880 GOTO 1340
890 CALL HCHAR(22,1,32,32)
900 CALL SCREEN(2)
910 FOR F=14 TO 16
920 CALL COLOR(F,2,2)
930 NEXT F
940 GOTO 3360
950 RETURN
960 CALL VCHAR(1,4,148,18)
970 CALL VCHAR(1,25,148,18)
980 CALL HCHAR(18,4,148,21)
990 FOR H=1 TO 6
1000 B=4+H
1010 CALL HCHAR(H,B,137)
1020 NEXT H
1030 CALL HCHAR(6,11,138,8)
1040 FOR H=1 TO 6
1050 B=25-H
1060 CALL HCHAR(H,B,136)
1070 NEXT H
1080 CALL VCHAR(7,19,139,9)
1090 CALL VCHAR(7,10,140,9)
1100 CALL VCHAR(9,12,140,6)
1110 CALL VCHAR(9,17,139,6)
1120 CALL HCHAR(8,13,138,4)
1130 CALL HCHAR(15,11,138)
1140 CALL HCHAR(15,18,138)
1150 CALL HCHAR(15,12,150)
1160 CALL HCHAR(15,17,149)
1170 CALL HCHAR(16,19,137)
1180 CALL HCHAR(16,10,136)
1190 CALL HCHAR(16,20,141)
1200 CALL HCHAR(16,9,141)
1210 CALL VCHAR(9,9,143,7)
1220 CALL VCHAR(9,20,143,7)
1230 CALL HCHAR(8,20,142)
1240 CALL HCHAR(8,9,142)
1250 CALL HCHAR(7,9,146)
1260 CALL HCHAR(7,20,144)
1270 CALL HCHAR(7,21,145)
1280 CALL HCHAR(6,21,151)
1290 CALL VCHAR(8,21,140,10)
1300 CALL HCHAR(7,8,147)
1310 CALL HCHAR(6,8,152)
1320 CALL VCHAR(8,8,139,10)
1330 GOSUB 2580
1340 START=RAU(HO,BR)
1350 ON START GOSUB 2020,232
0,2140,2080,2140,2200,2200,2
260,2020
1360 CALL SOUND(100,500,0,60
0,0)
1370 CALL SOUND(100,300,0,40
0,0)
1380 IF WE(HO,BR)<>0 THEN 14
10
1390 WERT=1
1400 G$="NICHTS"
1410 GOSUB 3740
1420 CALL HCHAR(16,17,WERT1)
1430 F$="SIE FINDEN:"
1440 FOR J=1 TO LEN(F$)
1450 CALL HCHAR(20,J+3,ASC(S
EG$(F$,J,1)))
1460 NEXT J
1470 IF HA<>HO THEN 1490
1480 IF BA=BR THEN 2910
1490 FOR J=1 TO LEN(G$)
1500 CALL HCHAR(22,J+3,ASC(S
EG$(G$,J,1)))
1510 NEXT J
1520 CALL KEY(0,K,S)
1530 CALL SOUND(-100,-5,24)
1540 IF S=0 THEN 1520
1550 CALL SOUND(100,500,0,60
0,0)
1560 ST$=CHR$(K)
1570 ENE=ENE-1
1580 IF ENE>0 THEN 1620
1590 CALL CLEAR
1600 PRINT "TUT MIR LEID": :
"SIE HABEN DIE GANZE ENERGIE
 VERBRAUCHT":"PUNKTE:";PU: :
"NOCHMAL?(J)"
1610 GOTO 3490
1620 IF ST$="L" THEN 1700
1630 IF ST$="R" THEN 1730
1640 IF ST$="V" THEN 1760
1650 IF ST$="Z" THEN 1670
1660 IF ST$<>"N" THEN 1520 E
LSE 1790
1670 GOSUB 890
1680 HO=HO+1
1690 GOTO 1330
1700 BR=BR-1
1710 GOSUB 890
1720 GOTO 1330
1730 BR=BR+1
```

```
1740 GOSUB 890
1750 GOTO 1330
1760 HO=HO-1
1770 GOSUB 890
1780 GOTO 1330
1790 IF WERT=1 THEN 1830
1800 IF WERT=6 THEN 1520
1810 IF WE(HO,BR)=0 THEN 183
0
1820 IF ZAHL(WERT)<2 THEN 18
50
1830 CALL SOUND(1000,110,0)
1840 GOTO 1520
1850 IF WERT1=128 THEN 1980
1860 IF WERT1=120 THEN 1950
1870 ZAHL(WERT)=ZAHL(WERT)+1
1880 WE(HO,BR)=0
1890 CALL HCHAR(16,17,32)
1900 FOR Y=600 TO 700 STEP 2
0
1910 CALL SOUND(-100,Y,0,Y+2
0,0)
1920 NEXT Y
1930 PU=PU+2
1940 GOTO 1520
1950 PU=PU+60
1960 WE(HO,BR)=0
1970 GOTO 1890
1980 ENE=ENE+10
1990 WE(HO,BR)=0
2000 PU=PU+2
2010 GOTO 1890
2020 G$="NICHTS"
2030 WERT=1
2040 V=16
2050 H=1
2060 GOSUB 2380
2070 RETURN
2080 G$="EINEN KELCH MIT ENE
RGIE"
2090 WERT=2
2100 V=9
2110 H=15
2120 GOSUB 2380
2130 RETURN
2140 G$="EINE BIBEL GEGEN GE
ISTER"
2150 WERT=3
2160 V=2
2170 H=15
2180 GOSUB 2380
2190 RETURN
2200 G$="EINE PILLE GEGEN IN
FEKTION"
2210 WERT=4
2220 V=8
2230 H=5
2240 GOSUB 2380
2250 RETURN
2260 G$="EINE TRUHE MIT GOLD
"
2270 WERT=5
2280 V=2
2290 H=7
2300 GOSUB 2380
2310 RETURN
2320 G$="EINEN HINWEIS"
2330 WERT=6
2340 V=7
2350 H=12
2360 GOSUB 2380
2370 RETURN
2380 FOR Y=14 TO 16
2390 CALL COLOR(Y,V,H)
2400 NEXT Y
2410 CALL SCREEN(H)
2420 GOSUB 2440
2430 RETURN
2440 EN$="ENERGIE"
2450 FOR E=1 TO LEN(EN$)
2460 CALL HCHAR(19,E+3,ASC(S
EG$(EN$,E,1)))
2470 NEXT E
2480 FOR E=1 TO LEN(STR$(ENE
))
2490 CALL HCHAR(19,11+E,ASC(
SEG$(STR$(ENE),E,1)))
2500 NEXT E
2510 FOR P=1 TO 6
2520 CALL HCHAR(19,16+P,ASC(
SEG$("PUNKTE",P,1)))
2530 NEXT P
2540 FOR P=1 TO LEN(STR$(PU)
)
2550 CALL HCHAR(19,24+P,ASC(
SEG$(STR$(PU),P,1)))
2560 NEXT P
2570 RETURN
2580 IF HO=0 THEN 2630
2590 IF HO=10 THEN 2650
2600 IF BR=10 THEN 2670
2610 IF BR=0 THEN 2690
2620 RETURN
2630 HO=9
2640 RETURN
2650 HO=1
2660 RETURN
2670 BR=1
2680 RETURN
2690 BR=9
2700 RETURN
2710 RANDOMIZE
2720 IF RND>.6 THEN 2820
2730 IF RND>.5 THEN 2780
2740 HA=HH
2750 BA=BH+1
2760 IF BA>9 THEN 2710
2770 RETURN
2780 HA=HH
2790 BA=BH-1
2800 IF BA<1 THEN 2710
2810 RETURN
```

```
2820 IF RND>.8 THEN 2890
2830 HA=HH-1
2840 IF HA<1 THEN 2710
2850 BA=BH
2860 RETURN
2870 HA=HH+1
2880 IF HA>9 THEN 2710
2890 BA=BH
2900 RETURN
2910 CALL CLEAR
2920 CALL SCREEN(15)
2930 PRINT "GRATULIERE!!": :
"DU HAST DEN AUSGANG GEFUNDE
N":"PUNKTE:";PU: :"BONUS:";E
NE*100: :"GESAMT:";PU+ENE*10
0
2940 ZAHL(4)=0
2950 ZAHL(3)=0
2960 RESTORE 3060
2970 FOR SOUND=1 TO 7
2980 READ D,F
2990 CALL SOUND(D,F,0,F+1,0)
3000 NEXT SOUND
3010 PRINT "NOCHMAL?(J)"
3020 CALL KEY(0,K,S)
3030 IF S=0 THEN 3020
3040 IF K=74 THEN 540
3050 END
3060 DATA -100,131,-100,123,
-100,123,-100,165,-100,147,1
00,175,1000,131
3070 REM ** ZEICHENDEFINITIO
N **
3080 CALL CHAR(96,"0000003307
87830")
3090 CALL CHAR(104,"7C6C446C
6C6C7C7C")
3100 CALL CHAR(112,"18181818
18001818")
3110 CALL CHAR(120,"00007EFF
81E7FFFF")
3120 CALL CHAR(128,"4444447C
1010107C")
3130 CALL CHAR(136,"0102040S
1020408O")
3140 CALL CHAR(137,"8040201O
0804020I")
3150 CALL CHAR(138,"0000000O
000000FF")
3160 CALL CHAR(139,"8080808O
8080808")
3170 CALL CHAR(140,"0101010I
0101010I")
3180 CALL CHAR(141,"81818181
818181FF")
3190 CALL CHAR(142,"FF818181
81818181")
3200 CALL CHAR(143,"0101010I
81818181")
3210 CALL CHAR(144,"0102040S
1020408I")
3220 CALL CHAR(145,"0305091I
21418101")
3230 CALL CHAR(146,"8040201O
0804020I")
3240 CALL CHAR(147,"C0A09088
84828180")
3250 CALL CHAR(148,"FFFFFFFF
FFFFFFFF")
3260 CALL CHAR(149,"8080808O
8080808FF")
3270 CALL CHAR(150,"0101010I
0101010FF")
3280 CALL CHAR(151,"0103050S
11214181")
3290 CALL CHAR(152,"80C0A090
88848281")
3300 CALL COLOR(9,16,1)
3310 CALL COLOR(10,2,1)
3320 CALL COLOR(11,13,1)
3330 CALL COLOR(12,16,1)
3340 CALL COLOR(13,5,1)
3350 RETURN
3360 RANDOMIZE
3370 REM ** GEFAHREN **
3380 IF RND<.79 THEN 950
3390 IF RND>=.8123 THEN 3600
3400 PRINT "IHNEN BEGEGNET E
IN GEIST"
3410 IF ZAHL(3)<1 THEN 3480
3420 PRINT :"ABER SIE HABEN
EINE BIBEL"
3430 FOR H=1 TO 500
3440 NEXT H
3450 CALL CLEAR
3460 ZAHL(3)=ZAHL(3)-1
3470 GOTO 960
3480 PRINT :"UND DA SIE KEIN
E BIBEL":"MEHR HABEN,WERDEN
SIE VON":"IHM GETOETET": :"N
OCHMAL?(J)":"PUNKTE:";PU
3490 RESTORE 3730
3500 FOR TM=1 TO 11
3510 READ DAUER,FREQUENZ
3520 CALL SOUND(DAUER,FREQUE
NZ,0,FREQUENZ+1,0)
3530 NEXT TM
3540 CALL KEY(0,K,S)
3550 IF S=0 THEN 3540
3560 ZAHL(3)=0
3570 ZAHL(4)=0
3580 IF K=74 THEN 540
3590 END
3600 PRINT "SIE WERDEN VON E
INEM VIRUS ":"INFIZIERT"
3610 IF ZAHL(4)<1 THEN 3680
3620 PRINT :"HABEN ABER NOCH
 EINE":"PILLE DIE SIE RETTET
"
3630 ZAHL(4)=ZAHL(4)-1
3640 FOR W=1 TO 500
3650 NEXT W
```

```
3660 CALL CLEAR
3670 GOTO 960
3680 PRINT :"UND WERDEN,DA S
IE KEINE":"PILLE MEHR HABEN,
VON IHM ":"GETOETET":"PUNKTE
:";PU
3690 PRINT : :"NOCHMAL?(J)":
 :
3700 ZAHL(4)=0
3710 ZAHL(3)=0
3720 GOTO 3490
3730 DATA 500,196,500,196,10
0,196,500,196,500,233,100,22
0,250,220,250,196,250,196,25
0,185,1000,196
3740 ON WERT GOTO 3750,3770,
3790,3810,3830,3850
3750 WERT1=32
3760 RETURN
3770 WERT1=128
3780 RETURN
3790 WERT1=104
3800 RETURN
3810 WERT1=96
3820 RETURN
3830 WERT1=120
3840 RETURN
3850 WERT1=112
3860 RETURN
```


Hard-and Software



Winterstraße 17
1000 Berlin 51
☎030/491 70 42

SUPER SKETCH™

Hardware:

Grafik-Tablett	98.- DM
CPS 99 32K RAM, 2*RS-232, 1*Centr.,	
Laufwerk + Controller	1698.- DM
Disklaufwerk TEAC 80 o. 40 Track	598.- DM
32 K ext.	295.- DM
32 K + Centr. ext.	395.- DM
Power Joystick 2St. f. TI	65.- DM
Akustikkoppler m. FTZ	289.- DM
Umrüstsatz Diskkontr. DS DD 80T.	155.- DM
Mini Assembler m. dtsch. Anl.	149.- DM

Software:

Mini Memory	M	295.- DM
Textgraf	D	79.- DM
Freddy	D	59.- DM

Fordern Sie unsere kostenlose Preisliste an
oder kommen Sie mal vorbei
öffnungszeiten: Mo-Fr 10-18h Sa 9-13h


```
100 ! **********************
110 ! *                    *
120 ! *      ECARTE        *
130 ! *   Copyright by     *
140 ! *Christian Weichmann *
150 ! *                    *
160 ! * Benoetigte Geraete *
170 ! *  TI99/4A  Konsole  *
180 ! *     Ext. Basic     *
190 ! *                    *
200 ! *  Speicherbelegung  *
210 ! *     8665 Bytes     *
220 ! **********************
225 !
230 RANDOMIZE
240 G=INT(RND*2)+1
250 DIM K(2,5),KF(4,8),S(32)
,P(2)
260 CALL VORBEREITUNG(G)
270 KFL=0
280 FOR L1=1 TO 4
290 FOR L2=1 TO 8
300 KF(L1,L2)=-1
310 NEXT L2
320 NEXT L1
330 FOR KT=1 TO 32
340 KN1=INT(RND*4+1)
350 KN2=INT(RND*8+1)
360 IF KF(KN1,KN2)=0 THEN 34
0
370 S(KT)=10*KN1+KN2
380 KF(KN1,KN2)=0
390 NEXT KT
400 Z=32
410 FOR KT=1 TO 5
420 FOR SP=3-G TO G STEP ((G
<3-G)+.5)*2
430 K(SP,KT)=S(Z)
440 Z=Z-1
450 CALL KARTE(-(SP=1)*S(Z+1
),5*(KT-1)+1,5+6*SP)
460 NEXT SP
470 NEXT KT
480 TK=S(Z)
490 T=INT(TK/10)
500 IF TK-T*10=8 THEN P(G)=P
(G)+1 :: KFL=-1 :: CALL BILD
SCHIRM(0,P(),0,0,G)
510 CALL KARTE(TK,3,14)
520 CALL KARTE(0,8,14)
530 FOR SZ=1 TO 21
540 IF INT(S(SZ)/10)=T THEN
S(SZ)=S(SZ)+100
550 NEXT SZ
560 AZ=32
570 SZ=21
580 FOR SP=1 TO 2
590 FOR I=1 TO 5
600 IF T=INT(K(SP,I)/10)THEN
 K(SP,I)=K(SP,I)+100
610 NEXT I
```

```
620 NEXT SP
630 CALL SORTIER(K(,))
640 FOR I=1 TO 5
650 CALL KARTE(K(1,I),(I-1)*
5+1,11)
660 NEXT I
670 M,STZ(1),STZ(2)=0
680 IF G=1 THEN CALL SPIEL(T
,P(),K(,),KFL,M):: M=M+(M=-1
):: CALL BILDSCHIRM(0,P(),0,
1-M,G)
690 IF M=-3 THEN GOTO 680
700 IF M THEN SP=2 :: GOTO 9
30
710 CALL KEY(0,M,Z):: IF Z=0
 THEN 710
720 IF M=65 AND K(1,1)=108+T
*10 AND NOT KFL THEN P(1)=P(
1)+1 :: KFL=NOT KFL :: CALL
BILDSCHIRM(0,P(),4,-1,G):: G
OTO 710
730 IF M=83 THEN SP=1 :: CAL
L BILDSCHIRM(0,P(),3+(G=1),-
1,G):: GOTO 930
740 IF M<>25 THEN 710
750 IF G=2 THEN CALL BILDSCH
IRM(0,P(),1,-1,G):: CALL SPI
EL(T,P(),K(,),KFL,M):: IF M
THEN SP=2 :: CALL BILDSCHIRM
(0,P(),-1,2,G):: GOTO 930
760 DISPLAY AT(12,(G-1)*21+1
)SIZE(7):"WIEVIEL"
770 IF G=2 THEN CALL KEY(0,A
,ST):: IF ST<1 OR A<48 OR A>
53 THEN 770 ELSE A=VAL(CHR$(
A)):: GOTO 790
780 CALL AUSTAUSCH(K(,),A)
790 DISPLAY AT(12,(G-2)*-21+
1)SIZE(7):A
800 ON G GOSUB 840,880
810 IF G=1 THEN CALL KEY(0,A
,ST):: IF ST<1 OR A<48 OR A>
53 THEN 810 ELSE A=VAL(CHR$(
A))
820 ON G GOSUB 830,840
830 CALL VCHAR(1,11,31,24)::
 CALL KARTE(0,18,14):: GOTO
630
840 FOR L=1 TO 5
850 IF SZ=0 THEN CALL KARTE(
-1,8,14):: RETURN
860 IF K(2,L)<100 THEN S(AZ)
=K(2,L):: AZ=AZ-1 :: K(2,L)=
S(SZ):: SZ=SZ-1
870 NEXT L :: RETURN
880 FOR L=1 TO A
890 IF SZ=0 THEN CALL KARTE(
-1,8,14):: RETURN
900 CALL SOUND(100,110,1)
910 CALL KEY(0,A,ST):: IF ST
<1 OR A<49 OR A>53 THEN 910
ELSE A=VAL(CHR$(A)):: S(AZ)=
K(1,A):: AZ=AZ-1 :: K(1,A)=S
(SZ):: SZ=SZ-1 :: CALL HCHAR
((A-1)*5+1,11,62)
920 NEXT L :: RETURN
930 A=(NOT(G-2))+2
940 IF A=2 THEN CALL AUSSPIE
L1(K(,),ASK):: GOTO 1050
950 CALL SOUND(100,110,1)
960 CALL KEY(0,KT,ST):: IF S
T=0 THEN 960
970 IF KT<49 OR KT>53 THEN 9
50
980 KT=VAL(CHR$(KT))
990 IF K(1,KT)=-1 THEN 950
1000 CALL KARTE(-1,5*(KT-1)+
1,11)
1010 ASK=K(1,KT)
1020 CALL KARTE(ASK,13,14)
1030 K(1,KT)=-1
1040 CALL AUSSPIEL2(T,K(,),A
SK,BGK):: GOTO 1150
1050 CALL SOUND(100,110,1)
1060 CALL KEY(0,KT,ST):: IF
ST=0 THEN 1060
1070 IF KT<49 OR KT>53 THEN
1050
1080 KT=VAL(CHR$(KT))
1090 IF K(1,KT)=-1 THEN 1050
1100 CALL KONTROLLIER(T,K(,)
,ASK,KT)
1110 CALL KARTE(-1,5*(KT-1)+
1,11)
1120 BGK=K(1,KT)
1130 CALL KARTE(BGK,13,14)
1140 K(1,KT)=-1
1150 IF INT(BGK/10)<>INT(ASK
/10)AND INT(BGK/10)<>10+T TH
EN BGK=-BGK
1160 A=-(ASK>BGK)*A-((NOT(A-
2))+2)*(ASK<BGK)
1170 STZ(A)=STZ(A)+1
1180 CALL KARTE(-1,13,14)
1190 CALL KARTE(0,5*(STZ(A)-
1)+1,A*12-4)
1200 IF STZ(1)+STZ(2)<5 THEN
 940
1210 IF STZ(SP)>2 THEN P(SP)
=P(SP)+1
1220 IF STZ(SP)=5 THEN P(SP)
=P(SP)+1
1230 IF STZ(SP)<3 THEN P(SP)
=P(SP)-2 :: SP=(NOT(SP-2))+2
1240 DISPLAY AT(20,(SP-1)*21
+1)SIZE(7):"SIEGER"
1250 CALL BILDSCHIRM(0,P(),0
,0,G)
1260 DISPLAY AT(21,22)SIZE(7
):"NOCH"
1270 DISPLAY AT(22,22)SIZE(7
):"EIN"
```

```
1280 DISPLAY AT(23,22)SIZE(7
):"SPIEL?"
1290 CALL KEY(0,A,ST):: IF S
T=0 THEN 1290
1300 IF A=78 THEN STOP
1310 IF A<>74 THEN 1280
1320 G=(NOT(G-2))+2
1330 CALL BILDSCHIRM(-1,P(),
0,0,G)
1340 GOTO 270
1350 SUB AUSTAUSCH(K(,),A)
1360 A=0 :: FOR L=1 TO 5
1370 IF K(2,L)<100 THEN A=A+
1
1380 NEXT L :: SUBEND
1390 SUB AUSSPIEL1(K(,),ASK)
1400 M=-1 :: KT=0
1410 FOR L=1 TO 5
1420 IF K(2,L)>M THEN M=K(2,
L):: KT=L
1430 NEXT L
1440 CALL KARTE(-1,(KT-1)*5+
1,17)
1450 CALL KARTE(K(2,KT),13,1
4)
1460 ASK=K(2,KT)
1470 K(2,KT)=-1
1480 SUBEND
1490 SUB AUSSPIEL2(T,K(,),AS
K,BGK)
1500 M=200 :: KT=0
1510 FOR L=1 TO 5
1520 IF K(2,L)<M AND K(2,L)>
ASK AND INT(K(2,L)/10)=INT(A
SK/10)THEN M=K(2,L):: KT=L
1530 NEXT L
1540 IF KT=0 THEN 1600
1550 CALL KARTE(-1,(KT-1)*5+
1,17)
1560 CALL KARTE(K(2,KT),13,1
4)
1570 BGK=K(2,KT)
1580 K(2,KT)=-1
1590 SUBEXIT
1600 FOR L=1 TO 5
1610 IF M>K(2,L)AND INT(K(2,
L)/10)=INT(ASK/10)THEN M=K(2
,L):: KT=L
1620 NEXT L
1630 IF KT>0 THEN 1550
1640 FOR L=1 TO 5
1650 IF K(2,L)<M AND INT(K(2
,L)/10)=10+T THEN M=K(2,L)::
 KT=L
1660 NEXT L
1670 IF KT>0 THEN 1550
1680 M=9
1690 FOR L=1 TO 5
1700 IF M-INT(M/10)*10>K(2,L
)-INT(K(2,L)/10)*10 AND K(2,
L)>-1 THEN M=K(2,L):: KT=L
1710 NEXT L :: GOTO 1550
1720 SUBEND
1730 SUB KONTROLLIER(T,K(,),
ASK,KTA)
1740 BGK=K(1,KTA)
1750 IF BGK>ASK AND INT(BGK/
10)=INT(ASK/10)THEN 1910
1760 M=200 :: KT=0 :: FOR L=
1 TO 5
1770 IF INT(K(1,L)/10)=INT(A
SK/10)AND K(1,L)<M AND K(1,L
)>ASK THEN KT=L :: M=K(1,L)
1780 NEXT L
1790 IF KT<>0 THEN 1900
1800 IF INT(BGK/10)=INT(ASK/
10)THEN 1910
1810 FOR L=1 TO 5
1820 IF INT(K(1,L)/10)=INT(A
SK/10)AND K(1,L)<M THEN M=K(
1,L):: KT=L
1830 NEXT L
1840 IF KT<>0 THEN 1900
1850 IF INT(BGK/10)=10+T THE
N 1910
1860 FOR L=1 TO 5
1870 IF M>K(1,L)AND INT(K(1,
L)/10)=10+T THEN M=K(1,L)::
KT=L
1880 NEXT L
1890 IF KT=0 THEN 1910
1900 KTA=KT
1910 SUBEND
1920 SUB VORBEREITUNG(G)
1930 FOR SZ=95 TO 143
1940 READ S$
1950 CALL CHAR(SZ,S$)
1960 NEXT SZ
1970 P(1),P(2)=0
1980 CALL BILDSCHIRM(-1,P(),
0,0,G)
1990 FOR I=8 TO 14
2000 CALL COLOR(I,2,16)
2010 NEXT I
2020 CALL COLOR(13,?,16)
2030 CALL COLOR(0,1,13)
2040 DATA AA55AA55AA55AA55,0
03F3F000000000003,030F0F0F0F0F
0F00,00F0F03030C0C000,00,000
F0F3030303000F,0F3030303000F0F
00
2050 DATA 00C0C0303030330C0,C
0303030330C0C000,000F0F3030303
000F,0F000000000F0F00,00C0C0
303030330F0,F03030C0C0000000
2060 DATA 003F3F000000000003,0
30C0C30303F3F00,00F0F03030C0
C000,000000000F0F000,000F0F
3030303003F,3F3030303030300
2070 DATA 00C0C0303030330F0,F
0303030303003000,003F3F0C0C0C
0C0F,0F0C0C0C0C3F3F00,00C0C0
```

```
303030303C0,C03030303C0C000
2080 DATA 003F3F0C0C0C0C0C,0
C0C0C0C0C3F3F00,00C0C0303030
3030,303030303C0C000,003030
303033333C,3C33333030303000
2090 DATA 003030C0C0000000,0
00000C0C0303000,0001030700F1F
3F7F,7F3F1F0F07030100,0080C0
E0F0F8FCFE,FEFCF8F0E0C08000
2100 DATA 000000183C3E3F3F,1
F0F0703010000000,000000183C7C
FCFC,F8F0E0C080000000,000103
070F1F3F3F,3F3D1901010103000
2110 DATA 0080C0E0F0F8FCFC,F
CBC9880808C000,0001030703011
397F,7F3911010101030000,0080C0
E0C0889CFE,FE9C8880808C000
2120 SUBEND
2130 SUB BILDSCHIRM(CF,P(),K
1,K2,G)
2140 IF CF THEN CALL CLEAR :
: CALL VCHAR(1,10,31,336)
2150 DISPLAY AT(1,1)SIZE(7):
"SPIELER"
2160 DISPLAY AT(1,22)SIZE(7)
:"SPIELER"
2170 DISPLAY AT(2,1)SIZE(2):
"#1"
2180 DISPLAY AT(2,22)SIZE(2)
:"#2"
2190 DISPLAY AT(4,1)SIZE(7):
"PUNKTE:"
2200 DISPLAY AT(4,22)SIZE(7)
:"PUNKTE:"
2210 DISPLAY AT(5,1)SIZE(7):
P(1)
2220 DISPLAY AT(5,22)SIZE(7)
:P(2)
2230 IF G=1 THEN DISPLAY AT(
7,1)SIZE(5):"GEBER" :: DISPL
AY AT(7,22)SIZE(7):"AUSSP."
2240 IF G=2 THEN DISPLAY AT(
7,22)SIZE(5):"GEBER" :: DISP
LAY AT(7,1)SIZE(7):"AUSSP."
2250 IF K1=1 OR K2=1 THEN DI
SPLAY AT(9,ABS((K1=1)+(K2=1)
*22))SIZE(7):"KARTE"
2260 IF K1=2 OR K2=2 THEN DI
SPLAY AT(9,ABS((K1=2)+(K2=2)
*22))SIZE(7):"SPIEL!"
2270 IF K1=3 OR K2=3 THEN DI
SPLAY AT(9,ABS((K1=3)+(K2=3)
*22))SIZE(7):"ICH" :: DISPLA
Y AT(10,ABS((K1=3)+(K2=3)*22
))SIZE(7):"SPIELE"
2280 IF K1=4 OR K2=4 THEN DI
SPLAY AT(9,ABS((K1=4)+(K2=4)
*22))SIZE(7):"MELDE" :: DISP
LAY AT(10,ABS((K1=4)+(K2=4)*
22))SIZE(7):"KOENIG!"
2290 IF K1=0 THEN DISPLAY AT
(9,1)SIZE(7):"" :: DISPLAY A
T(10,1)SIZE(7):""
2300 IF K2=0 THEN DISPLAY AT
(9,22)SIZE(7):"" :: DISPLAY
AT(10,22)SIZE(7):""
2310 SUBEND
2320 SUB KARTE(C,Z,S)
2330 S=S+2
2340 IF C<=0 THEN 2480
2350 FC=INT(C/10)-INT(C/100)
*10
2360 KC=C-FC*10+100*(C>100)
2370 FC=124+FC*4
2380 KC=92+KC*4
2390 CALL HCHAR(Z,S,FC)
2400 CALL HCHAR(Z,S+1,FC+2)
2410 CALL HCHAR(Z+1,S,FC+1)
2420 CALL HCHAR(Z+1,S+1,FC+3
)
2430 CALL HCHAR(Z+2,S,KC)
2440 CALL HCHAR(Z+2,S+1,KC+2
)
2450 CALL HCHAR(Z+3,S,KC+1)
2460 CALL HCHAR(Z+3,S+1,KC+3
)
2470 SUBEXIT
2480 IF C=0 THEN C=95 ELSE C
=31
2490 CALL VCHAR(Z,S,C,4)
2500 CALL VCHAR(Z,S+1,C,4)
2510 SUBEND
2520 SUB SORTIER(K(,))
2530 FOR I1=1 TO 4
2540 FOR I2=I1+1 TO 5
2550 IF K(1,I1)<K(1,I2)THEN
H=K(1,I2):: K(1,I2)=K(1,I1):
: K(1,I1)=H
2560 NEXT I2
2570 NEXT I1
2580 SUBEND
2590 SUB SPIEL(T,P(),K(,),KF
L,M)
2600 TZ,KZ,DZ,BZ=0
2610 FOR L=1 TO 5
2620 IF K(2,L)=100+T*10 AND(
NOT(KF2))THEN 2740
2630 KI=INT(K(2,L)/10)
2640 IF KI=10+T THEN TZ=TZ+1
 :: GOTO 2680
2650 IF K(2,L)-KI*10=8 THEN
KZ=KZ+1 :: GOTO 2680
2660 IF K(2,L)-KI*10=7 THEN
DZ=DZ+1 :: GOTO 2680
2670 IF K(2,L)-KI*10=6 THEN
BZ=BZ+1 :: GOTO 2680
2680 NEXT L
2690 IF TZ>=3 THEN M=-1 :: G
OTO 2730
2700 IF TZ>=2 AND KZ=2 THEN
M=-1 :: GOTO 2730
2710 IF KZ+DZ+BZ>=4 AND KZ+D
```

```
Z>=2 THEN M=-1 :: GOTO 2730
2720 M=0
2730 SUBEXIT
2740 IF KFL THEN P(1)=P(1)-1
2750 P(2)=P(2)+1
2760 M=-3
2770 KF2,KFL=-1
2780 SUBEND
```

BÖRSE

Verk. Centr. Schnittst. Kabel (neu) 60,– Drucke List. Bas + XBas. f. 10,– W. Bekcer Einst. Str. 11 8012 Ottobrunn

---

SUPER ANGEBOT !!! Kassettenrandvoll mit guten Pgm's ab Fr. 15,– NEU !! Protected entfernen, Listings von Ihren Pgm's, Kopien von Disks mit beschriebenem Sektor Null, etc. Gratisinfo bie J. Brönnimann, Ausmattstr. 5, CH-4132 Muttenz

---

TOPGRAFIK-Spiele für TI-99. Suche Software für Minimem. (Handbuch). Verk. Spiel-Module. Nach 8 Uhr Tel. 02841/57 971

Ti99/4A mit Zubehör komplett o. einzeln zu verkaufen. Exp.-Box, Karten: 48kRAM, RS232 Centronics, Controler 2 Floppylaufw., Speech-Synth. Ext. basic, Ed. Assembler, TI-Writer, viel Softw. VB 1500 kmpl. Tel.: 0511/75 76 24

Zu verkaufen:
2 TI 99/4A (davon 1 leicht defekt (ROM) mit Recorder + Kabel, Ext. Basic sowie 16 weitere Module zu verkaufen! Auch einzeln! Sowie Joystick etc.! Komplettpreis 1200,– DM o. Gebot! Tel. 07022/61950

Verk. TI99/4A + Ext. + 6 Module + Rec. Kabel + Joysticks: 478 Lippstadt, Bastionstr. 45, Hofknecht: (580,– DM)

Tausche Progr. in Ex.B auf Dsk. Suche RS232 Karte od. E/AMod. geg. Adv. Mod. m. ll Kas. u.a. A.Oest 06408/7650

Raum Langenfeld M/TG Lied des TI-Workshop sucht Anwender Prg. Tausch usw. Horst Nietowski 02173/15395

UNHEIMLICH – Der Weg ins Ungewisse: * THE* MASK*OF*TERROR* Das 3-tlg. GRAPHIC-SUPER ADVENTURE! Durch seine verschlungenen Pfade bis jetzt ungelöst. Jetzt auch in dt.! 3 Zehner im Brief an: H. Hambach, Regensb. -Str. 236, 8500 Nürnberg 30. IN XBASIC!

** ALIEN SOFTWARE PRESENTS: **
SYNTHY MASTER, der beste Musikynthes. für den TI+XB!! Der kann Alles: z.B. Vibrator, Geräusche, 3-, 2- und einf. Klänge Wiedeholg., Speichern, Laden, Drucken usw. Der Bildschirm wird zur Orgeltast! Für nur 15 DM bei H. Hambach Regensb.-Str. 236, 8500 Nürnberg 30

** ALIEN SOFTWARE PRESENTS **
Programme, die fast profess. Ansprüchen genügen! Feinste Menütechnik! Anwenderfrdl. !! Textv., Datei, Mathematik, Finanzberater, Tagebuch, HOT GAMES! Info 80 Pf.: H. Hambach, Regensb.-Str. 236, 85 Nürnberg

---

```
10 REM*********************
11 REM*                    *
12 REM* CATCH WITH CAR     *
13 REM*  Copyright by      *
14 REM* Dirk  Junghans     *
15 REM*                    *
16 REM*                    *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                    *
23 REM* Speicherbelegung *
24 REM*    6762 Bytes     *
25 REM*********************
26 REM
160 RANDOMIZE
170 REM DEFINE CHARACTERS
180 DATA 0003060606060606,04
CA4440206E8A8A,000000F08888E
424,7C80C060301B0F01,EAEA2A2
A2A2A2EA1,2464C48282828408
190 DATA 01,9ECC6071311C0F03
,304080000080
200 CALL CHAR(136,"427E5A181
85A7E42")
210 CALL CHAR(137,"00E7427E7
E42E700")
220 CALL CHAR(96,"02C2241818
244340")
230 CALL CHAR(97,"C324187E5A
663C42")
240 CALL CHAR(33,"3C18A5DBDB
A5183C")
250 CALL CHAR(104,"566A566A5
66A566A")
260 CALL CHAR(105,"00FFAA55A
A55FF00")
270 CALL CHAR(112,"18183CE7E
73C1818")
280 CALL CHAR(106,"808080808
0808080")
290 REM VORSPANN
300 RESTORE 180
310 FOR I=144 TO 152
320 READ A$
330 CALL CHAR(I,A$)
340 NEXT I
350 CALL CLEAR
360 CALL SCREEN(10)
370 PRINT TAB(8);"DIRK JUNGH
ANS ": : :TAB(9);"PRAESENTIE
RT":TAB(11);"FUER DEN"
380 PRINT : : :TAB(11);"TI99
/4A": :TAB(13);"VON": : : :
390 PRINT TAB(13);CHR$(144);
CHR$(145);CHR$(146):TAB(13);
CHR$(147);CHR$(148);CHR$(149
)
400 PRINT TAB(13);CHR$(150);
CHR$(151);CHR$(152): : : :
410 CALL SOUND(300,262,2)
420 CALL SOUND(900,349,2)
```

```
430 CALL SOUND(100,329,2)
440 CALL SOUND(100,349,2)
450 CALL SOUND(100,392,2)
460 CALL SOUND(300,440,2)
470 CALL SOUND(300,440,2)
480 CALL SOUND(300,440,2)
490 CALL SOUND(100,440,2)
500 CALL SOUND(100,466,2)
510 CALL SOUND(100,392,2)
520 CALL SOUND(450,440,2)
530 CALL SOUND(150,392,2)
540 CALL SOUND(300,349,2)
550 FOR I=1 TO 500
560 NEXT I
570 FOR I=1 TO 20
580 PRINT
590 NEXT I
600 CALL SCREEN(11)
610 Z=10
620 S=2
630 Q$="*******CATCH WITH CA
R**********"
640 GOSUB 2810
650 FOR I=1 TO 30
660 IF I>13 THEN 700
670 CALL SCREEN(I+1)
680 CALL SCREEN(I+3)
690 CALL SCREEN(I+2)
700 CALL SOUND(100,-7,I,440,
I,190,I,880,I)
710 NEXT I
720 CALL CLEAR
730 WAGEN=3
740 PUNKTE=100
750 R=1
760 ANJ=0
770 DD=0
780 CALL SCREEN(13)
790 CALL COLOR(1,11,15)
800 CALL COLOR(9,10,15)
810 CALL COLOR(10,2,15)
820 CALL COLOR(11,8,15)
830 CALL COLOR(14,6,15)
840 CALL VCHAR(1,31,106,24)
850 Q$="PUNKTE"
860 Z=3
870 S=32
880 GOSUB 2850
890 GOSUB 2890
900 CALL VCHAR(19,31,137,WAG
EN)
910 D=24
920 J=15
930 CALL HCHAR(D,J,136)
940 CALL KEY(2,K,ST)
950 IF ST=0 THEN 940
960 GOSUB 2690
970 CALL JOYST(2,K,ST)
980 IF K>0 THEN 1310
990 IF K<0 THEN 1450
1000 IF ST>0 THEN 1030
1010 IF ST<0 THEN 1170
1020 ON R GOTO 1030,1170,131
0,1450
1030 R=1
1040 L=136
1050 M=D
1060 N=J
1070 O=104
1080 D=D-1
1090 IF D<1 THEN 1590
1100 CALL GCHAR(D,J,U)
1110 IF U=32 THEN 1140
1120 GOSUB 3310
1130 GOTO 970
1140 CALL HCHAR(D,J,136)
1150 CALL HCHAR(D+1,J,104)
1160 GOTO 970
1170 R=2
1180 L=136
1190 M=D
1200 N=J
1210 O=104
1220 D=D+1
1230 IF D>24 THEN 1590
1240 CALL GCHAR(D,J,U)
1250 IF U=32 THEN 1280
1260 GOSUB 3310
1270 GOTO 970
1280 CALL HCHAR(D,J,136)
1290 CALL HCHAR(D-1,J,104)
1300 GOTO 970
1310 R=3
1320 L=137
1330 M=D
1340 N=J
1350 O=105
1360 J=J+1
1370 IF J>30 THEN 1590
1380 CALL GCHAR(D,J,U)
1390 IF U=32 THEN 1420
1400 GOSUB 3310
1410 GOTO 970
1420 CALL HCHAR(D,J,137)
1430 CALL HCHAR(D,J-1,105)
1440 GOTO 970
1450 R=4
1460 L=137
1470 M=D
1480 N=J
1490 O=105
1500 J=J-1
1510 IF J<1 THEN 1590
1520 CALL GCHAR(D,J,U)
1530 IF U=32 THEN 1560
1540 GOSUB 3310
1550 GOTO 970
1560 CALL HCHAR(D,J,137)
1570 CALL HCHAR(D,J+1,105)
1580 GOTO 970
1590 FOR I=1 TO 30
1600 CALL SOUND(-65,119,I,-5
```

```
,I,117,I,115,I)
1610 NEXT I
1620 GOSUB 3000
1630 IF D<1 THEN 1670
1640 IF D>24 THEN 1690
1650 IF J<1 THEN 1710
1660 IF J>30 THEN 1730
1670 D=1
1680 GOTO 1740
1690 D=24
1700 GOTO 1740
1710 J=1
1720 GOTO 1740
1730 J=30
1740 CALL HCHAR(D,J,32)
1750 D=INT(RND*24)+1
1760 J=INT(RND*30)+1
1770 CALL GCHAR(D,J,ZZ)
1780 IF ZZ<>32 THEN 1750
1790 IF J<25 THEN 1800 ELSE
1750
1800 R=3
1810 CALL HCHAR(D,J,137)
1820 GOTO 940
1830 CALL HCHAR(D,J,L)
1840 CALL HCHAR(M,N,O)
1850 CALL SOUND(100,440,1,55
0,1,660,1)
1860 CALL SOUND(100,990,1,10
00,1,1010,1)
1870 PUNKTE=PUNKTE+211
1880 GOSUB 2890
1890 GOTO 2690
1900 CALL HCHAR(D,J,L)
1910 CALL HCHAR(M,N,O)
1920 TON=220
1930 FOR I=1 TO 3
1940 CALL SOUND(100,TON,1)
1950 TON=TON+1
1960 NEXT I
1970 PUNKTE=PUNKTE+311
1980 GOSUB 2890
1990 GOTO 2690
2000 CALL HCHAR(D,J,L)
2010 CALL HCHAR(M,N,O)
2020 CALL SOUND(100,220,1)
2030 CALL SOUND(100,247,1)
2040 CALL SOUND(100,294,1)
2050 CALL SOUND(100,330,1)
2060 FOR I=1 TO 6
2070 DJ1=DJ1+1
2080 IF DJ1>7 THEN 2150
2090 X=INT(RND*24)+1
2100 Y=INT(RND*30)+1
2110 CALL GCHAR(X,Y,F)
2120 IF (F=105)+(F=104)THEN
2130 ELSE 2070
2130 CALL HCHAR(X,Y,32)
2140 CALL SOUND(100,440,1)
2150 DJ1=0
2160 NEXT I
2170 GOTO 2690
2180 CALL HCHAR(D,J,L)
2190 CALL HCHAR(M,N,O)
2200 CALL SOUND(100,262,1)
2210 CALL SOUND(100,524,1)
2220 CALL SOUND(100,392,1)
2230 CALL SOUND(100,440,1)
2240 CALL SOUND(100,344,1)
2250 PUNKTE=PUNKTE+90
2260 GOSUB 2890
2270 FOR I=1 TO 3
2280 DJ=DJ+1
2290 IF DJ>14 THEN 2360
2300 X=INT(RND*24)+1
2310 Y=INT(RND*30)+1
2320 CALL GCHAR(X,Y,F)
2330 IF (F=104)+(F=105)THEN
2340 ELSE 2280
2340 CALL HCHAR(X,Y,32)
2350 CALL SOUND(100,440,1)
2360 DJ=0
2370 NEXT I
2380 GOTO 2690
2390 CALL HCHAR(D,J,L)
2400 CALL HCHAR(M,N,O)
2410 CALL SOUND(100,-6,1,200
,1)
2420 ANJ=ANJ+1
2430 IF ANJ>15 THEN 3040
2440 FOR I=1 TO LEN(STR$(ANJ
))
2450 CALL HCHAR(1,30+I,ASC(S
EG$(STR$(ANJ),I,1)))
2460 NEXT I
2470 PUNKTE=PUNKTE-100
2480 GOSUB 2890
2490 GOTO 970
2500 DD=DD+1
2510 IF DD>1 THEN 2950
2520 RESTORE 2670
2530 FOR I=1 TO 13
2540 READ AB,CD
2550 CALL SOUND(AB,CD,1)
2560 NEXT I
2570 WAGEN=WAGEN+1
2580 CALL VCHAR(18+WAGEN,31,
137)
2590 FOR I=1 TO 25
2600 X=INT(RND*24)+1
2610 Y=INT(RND*30)+1
2620 CALL GCHAR(X,Y,U)
2630 IF (U=104)+(U=105)THEN
2640 ELSE 2600
2640 CALL HCHAR(X,Y,32)
2650 NEXT I
2660 GOTO 2950
2670 DATA 200,330,200,330,20
0,330,200,349,200,262,200,33
0,500,294,200,311,200,311,20
0,311,200,349,300,277
2680 DATA 500,247
```

```
2690 Z$="'a!p"
2700 Z=INT(RND*4)+1
2710 V=ASC(SEG$(Z$,Z,1))
2720 C=INT(RND*24)+1
2730 G=INT(RND*30)+2
2740 CALL GCHAR(C,G,E)
2750 IF E<>32 THEN 2720
2760 CALL SOUND(100,330,1)
2770 CALL SOUND(100,330,1)
2780 CALL SOUND(210,500,1)
2790 CALL HCHAR(C,G,V)
2800 RETURN
2810 FOR I=1 TO LEN(Q$)
2820 CALL HCHAR(Z,S+I,ASC(SE
G$(Q$,I,1)))
2830 NEXT I
2840 RETURN
2850 FOR I=1 TO LEN(Q$)
2860 CALL VCHAR(Z+I,S,ASC(SE
G$(Q$,I,1)))
2870 NEXT I
2880 RETURN
2890 IF PUNKTE<=0 THEN 2900
ELSE 2940
2900 PUNKTE=0
2910 CALL VCHAR(10,32,32,5)
2920 CALL VCHAR(10,32,48)
2930 GOTO 3000
2940 IF PUNKTE>1500 THEN 250
0
2950 CALL VCHAR(10,32,32,5)
2960 FOR I=1 TO LEN(STR$(PUN
KTE))
2970 CALL VCHAR(10+I,32,ASC(
SEG$(STR$(PUNKTE),I,1)))
2980 NEXT I
2990 RETURN
3000 WAGEN=WAGEN-1
3010 IF WAGEN<=0 THEN 3040
3020 CALL VCHAR(19+WAGEN,31,
32)
3030 RETURN
3040 CALL HCHAR(10,12,32,9)
3050 Q$="GAME OVER"
3060 Z=10
3070 S=12
3080 GOSUB 2810
3090 IF PUNKTE>HIGH THEN 311
0
3100 GOTO 3170
3110 HIGH=PUNKTE
3120 TON=110
3130 FOR I=1 TO 3
3140 CALL SOUND(100,TON,1)
3150 TON=TON+110
3160 NEXT I
3170 CALL HCHAR(15,5,32,14)
3180 Z=15
3190 S=5
3200 Q$="HIGHSCORE "&STR$(HI
GH)
3210 GOSUB 2810
3220 CALL HCHAR(20,5,32,18)
3230 Z=20
3240 S=5
3250 Q$="NEUES SPIEL  (J/N)"
3260 GOSUB 2810
3270 CALL KEY(3,K,ST)
3280 IF K=74 THEN 290
3290 IF K=78 THEN 3300 ELSE
3270
3300 STOP
3310 IF U=96 THEN 1830
3320 IF U=97 THEN 1900
3330 IF U=33 THEN 2000
3340 IF U=112 THEN 2180
3350 IF U=104 THEN 2390
3360 IF U=105 THEN 2390
3370 END
```


Die nächste
TI REVUE erscheint
am 28. Oktober

Disketten / Zubehör - Einführungsangebot

Stück		10	100	300
☐	MD 5¼" 1D Standard	3,05	2,85	2,55
☐	MD 5¼" 1D Quality	3,75	3,50	3,05
☐	MD 5¼" 2D Quality	4,50	4,20	3,75
☐	MD 5¼" 2D 96 TPI	6,60	5,85	5,20
☐	MD 5¼" HD Maxell	12,60	11,75	10,90
☐	MD 5¼" 1D Maxell	5,75	5,45	4,75
☐	MD 5¼" 2D Maxell	8,35	8,10	6,95
☐	MD 3½" 1D SONY	12,20	10,45	9,36
☐	MD 3½" 2D SONY	14,60	13,60	12,20
☐	MD 3½" 1D Maxell	8,90	7,20	-
☐	MD 3½" 2D Maxell	12,90	10,60	-
☐	Diskettenlocher	13,98	-	-
☐	V24-Kabel Kunden spezifisch	58,50	-	-

Anderes Zubehör auf Anfrage! Wir fertigen Ihnen jedes EDV-Kabel preiswert nach Ihren Wünschen! Alle Preise sind Nettopreise in DM/Stück.
Mindestbestellwert DM 60,-. Lieferung auf Rechnung ab DM 300,-.
Bei Neukunden und Beträgen unter 300,- DM Lieferung per NN oder VK.

Name

Straße

Ort

Datum

Unterschrift/Firmenstempel

Einfach Anzeige ausschneiden und als Bestellschein einsenden an:

MICHAEL MEISTER EDV-Marketing
Rheinstraße 47 · 7500 Karlsruhe 21
Tel. (07 21) 55 46 01 (Frau Staack)

```
10 REM*********************
11 REM*                   *
12 REM*    SKYDRIVER      *
13 REM*  Copyright  by    *
14 REM*   B.Jaehrling     *
15 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                   *
23 REM* Speicherbelegung *
24 REM*    9871 Bytes     *
25 REM*********************
26 REM
160 CALL CLEAR
170 CALL SCREEN(4)
180 REM ZEICHENDEFINITION
190 REM  WOLKEN
200 CALL CHAR(40,"000006070F
3B7F77")
210 CALL CHAR(41,"0078B052CE
FFF6DC")
220 CALL CHAR(42,"663D0C")
230 CALL CHAR(43,"70C87060")
240 CALL CHAR(152,"000000000
3020F07")
250 CALL CHAR(153,"0102386E3
F7FFF7F")
260 CALL CHAR(154,"0461F86BF
DFFBFFF")
270 CALL CHAR(155,"80E8FC914
0D8E87C")
280 CALL CHAR(156,"071F7BEE7
D0C02")
290 CALL CHAR(157,"F7B3FF5F5
3E17706")
300 CALL CHAR(158,"FFFFDFFF9
EDA84")
310 CALL CHAR(159,"FAB8ECB6E
641E420")
320 REM BAUEME
330 CALL CHAR(136,"1818183C7
EBD3C7E")
340 CALL CHAR(137,"000101030
6000103")
350 CALL CHAR(138,"FFFFFFFFF
FFFFFFF")
360 CALL CHAR(139,"008080C06
00080C0")
370 CALL CHAR(140,"03170E000
30F1F33")
380 CALL CHAR(141,"C0E87000C
0F0F8CC")
390 CALL CHAR(142,"27070F0F1
F3FFF7F")
400 CALL CHAR(143,"E4E0F0F0F
8FCFFFE")
410 CALL CHAR(144,"3C3C3C3C3
C3C7EFF")
420 REM  HUBSCHRAUBER
430 CALL CHAR(130,"000005060
F1703")
440 CALL CHAR(131,"00BF0003F
FFFFF07")
450 CALL CHAR(132,"00FE80E01
00888F0")
460 CALL CHAR(133,"030907")
470 CALL CHAR(134,"E048F0")
480 REM LANDEMARKE
490 CALL CHAR(112,"7FFFFF7F"
)
500 CALL CHAR(113,"FFFFFFFF"
)
510 CALL CHAR(115,"FEFFFFFE"
)
520 REM SPRINGER
530 CALL CHAR(120,"18187EBD3
C244281")
540 CALL CHAR(122,"3C7EFFC34
22418")
550 FOR I=3 TO 8
560 CALL COLOR(I,2,4)
570 NEXT I
580 PRINT TAB(7);"S K Y - D
I V E R": : : : :
590 PRINT TAB(14);"von": : :
 : :
600 PRINT TAB(7);"Burkhard J
aehrling": : : : : : : :
610 PRINT TAB(16);"Erkrath,
1984": :
620 CALL HCHAR(6,9,61,17)
630 DATA 300,350,200,330,200
,350,200,370,200,370,200,350
,200,370,200,390
640 DATA 200,410,350,390,200
,370,200,370,450,390
650 FOR K=1 TO 2
660 FOR I=1 TO 13
670 READ A,B
680 CALL SOUND(A-20,B,0,B-2,
5,B+2,5)
690 NEXT I
700 FOR L=1 TO 70
710 NEXT L
720 RESTORE 630
730 NEXT K
740 DATA 300,390,200,390,200
,350,200,350,200,290,200,290
,200,250,200,310,200,280,600
,250,200,200
750 RESTORE 740
760 FOR I=1 TO 10
770 READ A,B
780 CALL SOUND(A-30,B,0,B-2,
5,B+2,5)
790 NEXT I
800 RESTORE 630
810 FOR I=1 TO 23
820 READ A,B
830 CALL SOUND(A-30,B,0,B-2,
5,B+2,5)
840 NEXT I
```

```
850 CALL CLEAR
860 PRINT "   KURZE SPIELANL
EITUNG ?": : :
870 PRINT TAB(12);"[J;N]": :
 : : : : : : : :
880 CALL KEY(0,K,S)
890 IF S=0 THEN 880
900 IF K=78 THEN 930
910 IF K<>74 THEN 880
920 GOSUB 4060
930 CALL CLEAR
940 SP$="BITTE SPIELART WAEH
LEN:"
950 SA$="1; LEICHTERES SPIEL
"
960 SB$="2; ETWAS SCHWIERIGE
R"
970 FOR I=1 TO LEN(SP$)
980 CALL HCHAR(12,4+I,ASC(SE
G$(SP$,I,1)))
990 NEXT I
1000 FOR I=1 TO LEN(SA$)
1010 CALL HCHAR(15,6+I,ASC(S
EG$(SA$,I,1)))
1020 NEXT I
1030 FOR I=1 TO LEN(SB$)
1040 CALL HCHAR(17,6+I,ASC(S
EG$(SB$,I,1)))
1050 NEXT I
1060 CALL KEY(0,K,S)
1070 IF S=0 THEN 1060
1080 IF K=49 THEN 1120
1090 IF K<>50 THEN 1060
1100 SP=2
1110 GOTO 1130
1120 SP=1
1130 CALL CLEAR
1140 CALL SCREEN(2)
1150 FOR I=3 TO 7
1160 CALL COLOR(I,15,1)
1170 NEXT I
1180 CALL COLOR(2,15,8)
1190 CALL COLOR(16,15,8)
1200 CALL COLOR(14,13,8)
1210 CALL COLOR(15,14,4)
1220 CALL COLOR(13,7,8)
1230 CALL COLOR(12,2,8)
1240 CALL COLOR(11,7,4)
1250 S$="SCORE"
1260 FOR I=1 TO LEN(S$)
1270 CALL HCHAR(2,3+I,ASC(SE
G$(S$,I,1)))
1280 NEXT I
1290 M$="MEN"
1300 FOR I=1 TO LEN(M$)
1310 CALL HCHAR(2,15+I,ASC(S
EG$(M$,I,1)))
1320 NEXT I
1330 P=0
1340 MEN=5
1350 BZ=0
1360 GOSUB 3920
1370 GOSUB 4000
1380 IF SP=1 THEN 1410
1390 CALL COLOR(12,2,2)
1400 CALL HCHAR(2,23,121,7)
1410 M=0
1420 Y=0
1430 U=0
1440 L=0
1450 T=0
1460 PU=0
1470 R=0
1480 BL=0
1490 CALL COLOR(9,8,8)
1500 CALL HCHAR(4,1,97,608)
1510 CALL COLOR(10,4,4)
1520 CALL HCHAR(20,1,104,160
)
1530 FOR I=1 TO 100
1540 NEXT I
1550 RANDOMIZE
1560 REM WIND J/N
1570 IF SP=1 THEN 1810
1580 WM=INT(RND*3)+1
1590 IF WM=2 THEN 1630
1600 IF WM=3 THEN 1650
1610 WM$="WIND  O"
1620 GOTO 1660
1630 WM$="WIND <<"
1640 GOTO 1660
1650 WM$="WIND >>"
1660 FOR I=1 TO LEN(WM$)
1670 CALL HCHAR(2,22+I,ASC(S
EG$(WM$,I,1)))
1680 IF WM=1 THEN 1730
1690 IF WM=3 THEN 1720
1700 CALL SOUND(100,280,0)
1710 GOTO 1730
1720 CALL SOUND(100,600,0)
1730 NEXT I
1740 REM POSITION LANDEMARKE

1750 IF WM=1 THEN 1810
1760 IF WM=2 THEN 1790
1770 A=INT(RND*22)+8
1780 GOTO 1820
1790 A=INT(RND*23)+3
1800 GOTO 1820
1810 A=INT(RND*24)+4
1820 CALL HCHAR(24,A,113)
1830 CALL HCHAR(24,A-1,112)
1840 CALL HCHAR(24,A+1,115)
1850 REM POSITION BAEUME
1860 FOR I=1 TO 6
1870 B=18
1880 C=INT(RND*29)+2
1890 CALL GCHAR(B,C,Z)
1900 IF Z<>97 THEN 1870
1910 CALL GCHAR(B,C-1,Z)
1920 IF Z<>97 THEN 1870
1930 CALL GCHAR(B,C+1,Z)
```

```
1940 IF Z<>97 THEN 1870
1950 F=24
1960 CALL GCHAR(F,C,Q)
1970 IF Q<>104 THEN 1870
1980 CALL GCHAR(F,C-1,Q)
1990 IF Q<>104 THEN 1870
2000 CALL GCHAR(F,C+1,Q)
2010 IF Q<>104 THEN 1870
2020 CALL HCHAR(B,C,138)
2030 CALL HCHAR(B+1,C,138)
2040 CALL HCHAR(B+2,C,144)
2050 CALL HCHAR(B+1,C-1,142)
2060 CALL HCHAR(B+1,C+1,143)
2070 CALL HCHAR(B,C-1,140)
2080 CALL HCHAR(B,C+1,141)
2090 CALL HCHAR(B-1,C-1,137)
2100 CALL HCHAR(B-1,C,138)
2110 CALL HCHAR(B-1,C+1,139)
2120 CALL HCHAR(B-2,C,136)
2130 NEXT I
2140 REM WOLKEN
2150 FOR I=1 TO 9
2160 D=5
2170 E=INT(RND*28)+3
2180 CALL GCHAR(D,E,Z)
2190 IF Z<>97 THEN 2160
2200 CALL HCHAR(D,E,40)
2210 CALL HCHAR(D+1,E,42)
2220 CALL HCHAR(D,E+1,41)
2230 CALL HCHAR(D+1,E+1,43)
2240 NEXT I
2250 F=6
2260 G=INT(RND*21)+6
2270 CALL GCHAR(F,G,Z)
2280 IF Z<>97 THEN 2250
2290 CALL HCHAR(F,G,152)
2300 CALL HCHAR(F+1,G,156)
2310 CALL HCHAR(F,G+1,153)
2320 CALL HCHAR(F+1,G+1,157)
2330 CALL HCHAR(F,G+2,154)
2340 CALL HCHAR(F+1,G+2,158)
2350 CALL HCHAR(F,G+3,155)
2360 CALL HCHAR(F+1,G+3,159)
2370 REM FLUG
2380 CALL COLOR(12,2,8)
2390 CALL COLOR(13,7,4)
2400 FOR I=2 TO 45
2410 CALL COLOR(13,7,8)
2420 IF I>31 THEN 2440
2430 CALL SOUND(500,-8,0)
2440 IF I>=31 THEN 2660
2450 IF I>28 THEN 2510
2460 CALL GCHAR(7,I+4,W)
2470 CALL GCHAR(7,I+3,Z)
2480 IF W-Z=1 THEN 2590
2490 IF M<1 THEN 2510
2500 IF M<8 THEN 2700
2510 CALL HCHAR(7,I-1,97)
2520 CALL HCHAR(8,I,97)
2530 CALL HCHAR(7,I,130)
2540 CALL HCHAR(7,I+1,131)
2550 CALL HCHAR(8,I+1,133)
2560 CALL HCHAR(7,I+2,132)
2570 CALL HCHAR(8,I+2,134)
2580 GOTO 2710
2590 CALL HCHAR(7,I-1,97)
2600 CALL HCHAR(7,I,97)
2610 CALL HCHAR(8,I,97)
2620 CALL HCHAR(7,I+1,97)
2630 CALL HCHAR(8,I+1,97)
2640 M=M+1
2650 GOTO 2710
2660 CALL HCHAR(7,30,97,3)
2670 CALL HCHAR(8,29,97,4)
2680 Y=2
2690 GOTO 2710
2700 M=M+1
2710 FOR K=1 TO 10
2720 NEXT K
2730 IF Y=2 THEN 2800
2740 IF Y=1 THEN 2810
2750 CALL KEY(0,K,S)
2760 IF S=0 THEN 3400
2770 IF K<>32 THEN 3400
2780 CALL SOUND(50,600,0)
2790 Y=1
2800 IF U=0 THEN 3400
2810 IF WM=3 THEN 2870
2820 L=I-(U+R)
2830 IF SP=1 THEN 2920
2840 IF WM=1 THEN 2920
2850 IF L<3 THEN 2920
2860 GOTO 2890
2870 L=I-(U-R)
2880 IF L>31 THEN 2920
2890 R=R+1
2900 IF R<3 THEN 2930
2910 R=3
2920 BL=BL+1
2930 IF U>15 THEN 3400
2940 IF U=15 THEN 3130
2950 IF U>12 THEN 3050
2960 CALL VCHAR(7+U,L,97)
2970 IF BL>2 THEN 3060
2980 IF WM=3 THEN 3020
2990 CALL VCHAR(7+U,L+1,97)
3000 CALL VCHAR(8+U,L+1,97)
3010 GOTO 3060
3020 CALL VCHAR(7+U,L-1,97)
3030 CALL VCHAR(8+U,L-1,97)
3040 GOTO 3060
3050 CALL VCHAR(7+U,L,104)
3060 CALL VCHAR(8+U,L,122)
3070 CALL VCHAR(9+U,L,120)
3080 IF 9+U+1>19 THEN 3110
3090 CALL GCHAR(9+U+1,L,V)
3100 IF V<>97 THEN 3260
3110 U=U+1
3120 GOTO 3400
3130 CALL VCHAR(22,L,104)
3140 CALL GCHAR(24,L,X)
3150 IF X=104 THEN 3340
```

```
3160 IF X=113 THEN 3190
3170 PU=1
3180 GOTO 3360
3190 PU=3
3200 CALL SOUND(100,250,0)
3210 CALL SOUND(100,280,0)
3220 CALL SOUND(100,320,0)
3230 CALL SOUND(300,430,0)
3240 U=U+1
3250 GOTO 3400
3260 CALL HCHAR(8+U,L,97)
3270 CALL SOUND(100,210,1)
3280 CALL HCHAR(9+U,L,97)
3290 CALL SOUND(100,180,1)
3300 CALL HCHAR(21,L,120)
3310 CALL SOUND(250,150,1)
3320 U=16
3330 GOTO 3380
3340 CALL SOUND(200,180,0)
3350 GOTO 3380
3360 CALL SOUND(100,250,0)
3370 CALL SOUND(250,425,0)
3380 T=1
3390 U=U+1
3400 NEXT I
3410 IF U>0 THEN 3430
3420 T=1
3430 IF PU=0 THEN 3490
3440 IF PU=1 THEN 3470
3450 P=P+3
3460 GOTO 3480
3470 P=P+1
3480 GOSUB 3910
3490 IF T=0 THEN 3520
3500 MEN=MEN-1
3510 GOSUB 4000
3520 IF BZ=1 THEN 3650
3530 IF P<30 THEN 3650
3540 BZ=1
3550 FOR I=1 TO 5
3560 BO$="BONUS SPRINGER"
3570 FOR K=1 TO LEN(BO$)
3580 CALL HCHAR(10,9+K,ASC(S
EG$(BO$,K,1)))
3590 NEXT K
3600 CALL SOUND(-200,600,0,5
00,0,700,0)
3610 CALL HCHAR(10,9,97,15)
3620 NEXT I
3630 MEN=MEN+1
3640 GOSUB 4000
3650 IF MEN=0 THEN 3670
3660 GOTO 1380
3670 CALL CLEAR
3680 CALL SCREEN(4)
3690 FOR I=3 TO 8
3700 CALL COLOR(I,2,4)
3710 NEXT I
3720 E$="SIE HABEN "&STR$(P)
&" PUNKTE"
3730 FOR I=1 TO LEN(E$)
3740 CALL HCHAR(14,4+I,ASC(S
EG$(E$,I,1)))
3750 NEXT I
3760 FOR I=1 TO 300
3770 NEXT I
3780 N$="NOCH EIN SPIEL ? [J
;N]"
3790 FOR I=1 TO LEN(N$)
3800 CALL HCHAR(20,4+I,ASC(S
EG$(N$,I,1)))
3810 NEXT I
3820 CALL KEY(0,K,S)
3830 IF S=0 THEN 3820
3840 IF K=74 THEN 930
3850 IF K<>78 THEN 3820
3860 CALL CLEAR
3870 FOR I=1 TO 5
3880 PRINT "      AUF WIEDER
SEHEN": : : : : : : : : :
: : : : : : : : :
3890 NEXT I
3900 END
3910 REM ZAEHLER PUNKTE
3920 P$=STR$(P)
3930 IF LEN(P$)=3 THEN 3960
3940 P$="0"&P$
3950 GOTO 3930
3960 FOR I=1 TO LEN(P$)
3970 CALL HCHAR(2,10+I,ASC(S
EG$(P$,I,1)))
3980 NEXT I
3990 RETURN
4000 REM ZAEHLER MEN
4010 ME$=STR$(MEN)
4020 CALL HCHAR(2,20,ASC(ME$
))
4030 FOR I=1 TO 100
4040 NEXT I
4050 RETURN
4060 CALL CLEAR
4070 PRINT "EIN HUBSCHRAUBER
 MIT 5 FALL-":"SCHIRMSPRINGE
RN FLIEGT UEBER":"EINE LANDS
CHAFT.": :
4080 PRINT "DURCH DRUECKEN D
ER SPACE-":"TASTE WIRD DER S
PRUNG AUS-":"GELOEST.": :
4090 PRINT "TRIFFT DER SPRIN
GER DIE":"LANDEMARKE NICHT W
IRD EIN":"SPRINGER ABGEZOGEN
.": :
4100 PRINT "TRIFFT ER SIE AM
 RAND WIRD":"EBENFALLS EIN S
PRINGER":"ABGEZOGEN;":"SIE E
RHALTEN ABER 1 PUNKT.": :
4110 PRINT "LANDET ER AUF DE
R MITTE DER":"LANDEMARKE ERH
ALTEN SIE":"3 PUNKTE."
4120 PRINT "DIE ANZAHL DER S
PRINGER":"AENDERT SICH NICHT
.":
```

```
4130 PRINT TAB(24);"ENTER"
4140 CALL KEY(3,K,S)
4150 IF S=0 THEN 4140
4160 IF K<>13 THEN 4140
4170 CALL CLEAR
4180 PRINT "WENN SIE 30 PUNK
TE ERREICHEN":"ERHALTEN SIE
EINEN SPRINGER":"ZUSAETZLICH
": : : : : : : : : :
4190 PRINT TAB(24);"ENTER"
4200 CALL KEY(3,K,S)
4210 IF S=0 THEN 4200
4220 IF K<>13 THEN 4200
4230 CALL CLEAR
4240 PRINT "SIE KOENNEN ZWIS
CHEN ZWEI":"SPIELVARIANTEN W
AEHLEN.": : : :
4250 PRINT "SPIELART 1:": :
4260 PRINT "ABSPRUNG BEI WIN
DSTILLE.": : :
4270 PRINT "SPIELART 2:": :
4280 PRINT "HIERBEI SIND DIE
 WINDVER-":"HAELTNISSE ZU BE
ACHTEN.":"SIE WERDEN VOR DEM
 START DES":"HUBSCHRAUBERS A
NGEZEIGT.": : : :
4290 PRINT "VIEL VERGNUEGEN
BEIM SPIEL": :
4300 PRINT TAB(24);"ENTER"
4310 CALL KEY(3,K,S)
4320 IF S=0 THEN 4310
4330 IF K<>13 THEN 4310
4340 RETURN
```

BÖRSE

Suche Kontakte zu TI99/4A Besitzer (auch Wissensch. Anwend.) im Raum Göttingen / Kass. WHXBREW 0551/71785

Suche Monitor für TI99/4A U.S.-Bauart am liebsten Farbe mit NTSC Anschluß. Um 400 DM. 0551/71785

Verk. TI+Ex-Basic-Programme. Info g. Rückporto. B. Knedel, Tulpengasse 16, 3171 Weyhausen Tel. 05362/71187

Suche E/A-Modul + 32K Erw. f. TI-99/4A. Marco Bock, Moselstr. 9, 2805 Stuhr I, Tel. 0421/89026

Verk. TI99/4A + Rekorder + Kabel +12 Steckmodule (Parsec, Schach, ext. Basic, Invaders) + Liter. + Software auf Cassette. Interessenten melden bei: Axel Knaus, Arminstr. 7, Stgt. 1, Tel.: 0711/608566. Topzustand! VB 600,– DM

Suche TM990/100 MA Mikro-Computer Anwende-Handbuch! Peter Heinz 06192/6066

Verkaufe TI99/4A mit X Basic, Software, Joysticks, Rekorder und Literatur für 550 DM. Roland Lux, Meisenweg 28, 8264 Waldkraiburg Tel. 08638/674 35

```
10 REM********************
11 REM*                  *
12 REM*    TI-MUEHLE     *
13 REM*  Copyright  by   *
14 REM* Juergen Bruehne  *
15 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                  *
23 REM* Speicherbelegung *
24 REM*   11294 Bytes    *
25 REM********************
26 REM
110 REM  BEI VERWENDUNG VON
EINEM  SCHWARZWEISSFERNSEHER
 IST ES SINNVOLL IN ZEILE 85
0 AUS DER 6 EINE 16 ZU MACHE
N!
120 REM ALLE REM-ZEILEN KOEN
NEN WEGGELASSEN WERDEN
130 CALL SCREEN(2)
140 CALL CLEAR
150 PRINT "  JUERGEN BRUEHNE
": :"  AM TANNENGARTEN 15":
:"  3340 AROLSEN-MASSENHAUSE
N": : : : :"  STELLT VOR;":"
  -----------": : :
160 FOR X=1 TO 15
170 CALL COLOR(X,16,2)
180 NEXT X
190 FOR X=1 TO 500
200 NEXT X
210 CALL SCREEN(14)
220 PRINT : : : : : : : : :
: : : : : : : : : : : : : :
:"  IN DER GRUNDWERSION": :
: : : : : : : : : : : : : :
: : : :
230 PRINT : : :"* DREIDIMMEN
SIONALE-MUEHLE ":"  --------
----------------": : :" AUF
DEM:": :" TI-99/4A": :" VON
TEXAS INSTRUMENTS": :,"ENTER
 DRUECKEN":
240 CALL KEY(0,KE,ST)
250 IF KE<>13 THEN 240
260 PRINT : : :"  BENOETIGEN
 SIE": :"  DIE SPIELREGELN ?
 (J/N)": :
270 CALL KEY(0,KE,ST)
280 IF ST=0 THEN 270
290 IF KE=78 THEN 520
300 CALL CLEAR
310 PRINT ".........SPIELREGE
LN":".........----------": :
 :"SINN DIESES SPIELES IST E
S,": :"SO VIELE MUEHLEN WIE
": :
320 PRINT "MOEGLICH ZU ERSCH
LIESSEN !": :"(ES GIBT 76 MO
EGLICHKEITEN":"ZU EINER MUEH
```

```
LE ZU KOMMEN)": : : :,"ENTER
 DRUECKEN": : : :
330 CALL KEY(0,KE,ST)
340 IF ST=0 THEN 330
350 PRINT " EINE MUEHLE BEST
EHT AUS": :" 4 SPIELSTEINEN
,DIE IN": :" EINER REIHE ANG
EORDNET ": :" SEIN MUESSEN."
: :
360 CALL KEY(0,KE,ST)
370 IF ST=0 THEN 360
380 PRINT : : : :" ALSO,WENN
 MAN VIER STEINE": :" (DER G
LEICHEN FARBE)": :" MIT EINE
M LINEAL DURCH": :" IHRE MIT
TELPUNKTE": :
390 PRINT " VERBINDEN KANN ,
HAT MAN": :" EINE MUEHLE": :
" AM ENDE DES SPIELES ": :"
WERTET IHR 'TI' DIE MUEHLEN"
: :
400 PRINT "AUS !": :" ICH WU
ENSCHE IHNEN ": :" EIN SPANN
ENDES SPIEL": :,"ENTER DRUEC
KEN"
410 CALL KEY(0,KE,ST)
420 IF KE<>13 THEN 410
430 CALL CLEAR
440 PRINT "    HALT !": : :"
 BALD HAETTE ICH": :" ES VER
GESSEN !": : : :" FALLS SIE
KEINEN": :" MITSPIELER HABEN
 ,": :
450 PRINT " DANN KOENNEN SIE
 AUCH GEGEN": :" MICH SPIELE
N.": :" DAZU GEBEN SIE NUR '
TI' ": :" ANSTELLE DES ZWEIT
EN"
460 PRINT "SPIELERS EIN.": :
 :"VIEL GLUECK IHR": :,"TI-9
9/4A": :"ENTER DRUECKEN": :
470 CALL KEY(0,KE,ST)
480 IF KE<>13 THEN 470
490 FOR X=1 TO 15
500 CALL COLOR(X,2,1)
510 NEXT X
520 CALL CLEAR
530 PRINT "BITTE NAMEN EINGE
BEN !": : :
540 INPUT "SPIELER 1: ":A$
550 INPUT "SPIELER 2: ":B$
560 FOR X=1 TO 32
570 CALL VCHAR(1,X,32,24)
580 NEXT X
590 CALL SCREEN(4)
600 D(4)=T
610 REM DEFINIERUNGEN !
620 CALL CHAR(130,"FFFFFFFFF
FFFFFFF")
630 CALL CHAR(140,"187E7EFFF
F7E7E18")
640 CALL CHAR(150,"187E7EFFF
F7E7E18")
650 CALL COLOR(13,2,13)
660 CALL COLOR(14,7,1)
670 CALL COLOR(15,6,1)
680 CALL COLOR(12,13,13)
690 C$="WEITER MIT"
700 D$=" (SPACE)"
710 E$="MOECHTEN SIE"
720 F$="ES NOCHMAL VERSUCHEN
 ?(J/N)"
730 Q=140
740 R=150
750 T=130
760 CALL CHAR(131,"0000001818
000000")
770 CALL CHAR(141,"E0207
A0E090F")
780 CALL HCHAR(1,1,141,768)
790 H$="MUEHLEN.......MUEHLE
N"
800 REM SPIELAUFBAU
810 CALL CLEAR
820 CALL SCREEN(4)
830 PRINT "  ";A$: :"  ";B$:
840 CALL HCHAR(8,1,120,320)
850 FOR X=1 TO 16
860 READ S
870 READ U
880 CALL VCHAR(18,3+ZA,X+64)
890 CALL VCHAR(S,3+ZA,131,U)
900 ZA=ZA+2
910 IF ZA=8 THEN 940
920 IF ZA=15 THEN 940
930 IF ZA<>22 THEN 960
940 RESTORE 4600
950 ZA=ZA-1
960 NEXT X
970 FOR X=1 TO 16
980 READ S
990 READ Z
1000 CALL VCHAR(S,Z,T,4)
1010 NEXT X
1020 REM ABFRAGE DES BESTIM-
        MUNGSORTES (WIRD
1030 REM MIT DER TASTATUR
1040 REM ABGEFRAGT)
1050 ZA=0
1060 CALL HCHAR(21,3,Q)
1070 CALL HCHAR(23,3,R)
1080 CALL HCHAR(23,24,120)
1090 CALL HCHAR(23,26,120)
1100 CALL HCHAR(21,24,94)
1110 CALL HCHAR(21,26,63)
1120 Y=Q
1130 IF Y<>R THEN 1150
1140 IF B$="TI" THEN 4210
1150 CALL KEY(0,KE,ST)
1160 IF ST=0 THEN 1130
1170 IF KE<65 THEN 1130
1180 IF KE>80 THEN 1130
```

```
1190 ON KE-64 GOTO 1200,1300
,1410,1500,1590,1680,1770,18
60,1950,2040,2130,2220,2310,
2400,2490,2580

1200 CALL GCHAR(13,3,A(4))
1210 IF A(4)<>T THEN 1130
1220 REM SETZEN DER SPIEL-
       STEINE
1230 FOR X=1 TO 4
1240 CALL GCHAR(17-X,3,A(X))
1250 IF A(X)<>T THEN 1280
1260 CALL HCHAR(17-X,3,Y)
1270 GOTO 2670
1280 NEXT X
1290 GOTO 2670
1300 CALL GCHAR(11,5,B(4))
1310 IF B(4)<>T THEN 1130
1320 FOR X=1 TO 4
1330 CALL GCHAR(15-X,5,B(X))
1340 IF B(X)<>T THEN 1370
1350 CALL HCHAR(15-X,5,Y)
1360 GOTO 2670
1370 NEXT X
1380 CALL GCHAR(9,7,C(4))
1390 IF C(4)<>T THEN 1130
1400 GOTO 2670
1410 CALL GCHAR(9,7,C(4))
1420 IF C(4)<>T THEN 1130
1430 FOR X=1 TO 4
1440 CALL GCHAR(13-X,7,C(X))
1450 IF C(X)<>T THEN 1480
1460 CALL HCHAR(13-X,7,Y)
1470 GOTO 2670
1480 NEXT X
1490 GOTO 2670
1500 CALL GCHAR(7,9,D(4))
1510 IF D(4)<>T THEN 1130
1520 FOR X=1 TO 4
1530 CALL GCHAR(11-X,9,D(X))
1540 IF D(X)<>T THEN 1570
1550 CALL HCHAR(11-X,9,Y)
1560 GOTO 2670
1570 NEXT X
1580 GOTO 2670
1590 CALL GCHAR(13,10,E(4))
1600 IF E(4)<>T THEN 1130
1610 FOR X=1 TO 4
1620 CALL GCHAR(17-X,10,E(X)
)
1630 IF E(X)<>T THEN 1660
1640 CALL HCHAR(17-X,10,Y)
1650 GOTO 2670
1660 NEXT X
1670 GOTO 2670
1680 CALL GCHAR(11,12,F(4))
1690 IF F(4)<>T THEN 1130
1700 FOR X=1 TO 4
1710 CALL GCHAR(15-X,12,F(X)
)
1720 IF F(X)<>T THEN 1750
1730 CALL HCHAR(15-X,12,Y)
1740 GOTO 2670
1750 NEXT X
1760 GOTO 2670
1770 CALL GCHAR(9,14,G(4))

1780 IF G(4)<>T THEN 1130
1790 FOR X=1 TO 4
1800 CALL GCHAR(13-X,14,G(X)
)
1810 IF G(X)<>T THEN 1840
1820 CALL HCHAR(13-X,14,Y)
1830 GOTO 2670
1840 NEXT X
1850 GOTO 2670
1860 CALL GCHAR(7,16,H(4))
1870 IF H(4)<>T THEN 1130
1880 FOR X=1 TO 4
1890 CALL GCHAR(11-X,16,H(X)
)
1900 IF H(X)<>T THEN 1930
1910 CALL HCHAR(11-X,16,Y)
1920 GOTO 2670
1930 NEXT X
1940 GOTO 2670
1950 CALL GCHAR(13,17,I(4))
1960 IF I(4)<>T THEN 1130
1970 FOR X=1 TO 4
1980 CALL GCHAR(17-X,17,I(X)
)
1990 IF I(X)<>T THEN 2020
2000 CALL HCHAR(17-X,17,Y)
2010 GOTO 2670
2020 NEXT X
2030 GOTO 2670
2040 CALL GCHAR(11,19,J(4))
2050 IF J(4)<>T THEN 1130
2060 FOR X=1 TO 4
2070 CALL GCHAR(15-X,19,J(X)
)
2080 IF J(X)<>T THEN 2110
2090 CALL HCHAR(15-X,19,Y)
2100 GOTO 2670
2110 NEXT X
2120 GOTO 2670
2130 CALL GCHAR(9,21,K(4))
2140 IF K(4)<>T THEN 1130
2150 FOR X=1 TO 4
2160 CALL GCHAR(13-X,21,K(X)
)
2170 IF K(X)<>T THEN 2200
2180 CALL HCHAR(13-X,21,Y)
2190 GOTO 2670
2200 NEXT X
2210 GOTO 2670
2220 CALL GCHAR(7,23,L(4))
2230 IF L(4)<>T THEN 1130
2240 FOR X=1 TO 4
2250 CALL GCHAR(11-X,23,L(X)
)
2260 IF L(X)<>T THEN 2290
```

```
2270 CALL HCHAR(11-X,23,Y)
2280 GOTO 2670
2290 NEXT X
2300 GOTO 2670
2310 CALL GCHAR(13,24,M(4))
2320 IF M(4)<>T THEN 1130
2330 FOR X=1 TO 4
2340 CALL GCHAR(17-X,24,M(X)
)
2350 IF M(X)<>T THEN 2380
2360 CALL HCHAR(17-X,24,Y)
2370 GOTO 2670
2380 NEXT X
2390 GOTO 2670
2400 CALL GCHAR(11,26,N(4))
2410 IF N(4)<>T THEN 1130
2420 FOR X=1 TO 4
2430 CALL GCHAR(15-X,26,N(X)
)
2440 IF N(X)<>T THEN 2470
2450 CALL HCHAR(15-X,26,Y)
2460 GOTO 2670
2470 NEXT X
2480 GOTO 2670
2490 CALL GCHAR(9,28,O(4))
2500 IF O(4)<>T THEN 1130
2510 FOR X=1 TO 4
2520 CALL GCHAR(13-X,28,O(X)
)
2530 IF O(X)<>T THEN 2560
2540 CALL HCHAR(13-X,28,Y)
2550 GOTO 2670
2560 NEXT X
2570 GOTO 2670
2580 CALL GCHAR(7,30,P(4))
2590 IF P(4)<>T THEN 1130
2600 FOR X=1 TO 4
2610 CALL GCHAR(11-X,30,P(X)
)
2620 IF P(X)<>T THEN 2650
2630 CALL HCHAR(11-X,30,Y)
2640 GOTO 2670
2650 NEXT X
2660 GOTO 2670
2670 ZA=ZA+1
2680 CALL SOUND(-8,560,1)
2690 IF ZA=64 THEN 2780
2700 IF Y<>R THEN 1080
2710 Y=R
2720 CALL HCHAR(21,24,120)
2730 CALL HCHAR(21,26,120)
2740 CALL HCHAR(23,24,94)
2750 CALL HCHAR(23,26,63)
2760 IF B$="TI" THEN 4210
2770 GOTO 1130
2780 CALL HCHAR(23,24,130,3)
2790 GOSUB 3130
2800 Z=21
2810 S=20
2820 G$=C$
2830 GOSUB 3080
2840 Z=23
2850 G$=D$
2860 GOSUB 3080
2870 CALL KEY(0,KE,ST)
2880 IF KE<>32 THEN 2870
2890 CALL HCHAR(1,1,141,768)
2900 Z=13
2910 S=3
2920 G$=E$
2930 GOSUB 3080
2940 Z=15
2950 G$=F$
2960 GOSUB 3080
2970 CALL KEY(0,KE,ST)
2980 IF ST=0 THEN 2970
2990 IF KE<>74 THEN 3020
3000 RESTORE
3010 GOTO 520
3020 CALL CLEAR
3030 PRINT ".....AUF WIEDERS
EHEN!": : : : : : : :
3040 FOR X=1 TO 200
3050 NEXT X
3060 END
3070 REM   SCHREIBEN DER
           STRINGS
3080 FOR X=1 TO LEN(G$)
3090 CALL HCHAR(Z,S+X-1,ASC(
SEG$(G$,X,1)))
3100 NEXT X
3110 RETURN
3120 REM *MUEHLENAUSWERTUNG*
3130 CALL GCHAR(13,3,A(4))
3140 CALL GCHAR(11,5,B(4))
3150 CALL GCHAR(9,7,C(4))
3160 CALL GCHAR(7,9,D(4))
3170 CALL GCHAR(13,10,E(4))
3180 CALL GCHAR(11,12,F(4))
3190 CALL GCHAR(9,14,G(4))
3200 CALL GCHAR(7,16,H(4))
3210 CALL GCHAR(13,17,I(4))
3220 CALL GCHAR(11,19,J(4))
3230 CALL GCHAR(9,21,K(4))
3240 CALL GCHAR(7,23,L(4))
3250 CALL GCHAR(13,24,M(4))
3260 CALL GCHAR(11,26,N(4))
3270 CALL GCHAR(9,28,O(4))
3280 CALL GCHAR(7,30,P(4))
3290 SW=560
3300 ZZ=0
3310 FOR X=1 TO 6
3320 READ AA,BB,CC,DD
3330 IF A(AA)+B(BB)+C(CC)+D(
DD)<>SW THEN 3350
3340 ZZ=ZZ+1
3350 IF E(AA)+F(BB)+G(CC)+H(
DD)<>SW THEN 3370
3360 ZZ=ZZ+1
3370 IF I(AA)+J(BB)+K(CC)+L(
```

```
DD)<>SW THEN 3390
3380 ZZ=ZZ+1
3390 IF M(AA)+N(BB)+O(CC)+P(
DD)<>SW THEN 3410
3400 ZZ=ZZ+1
3410 IF A(AA)+E(BB)+I(CC)+M(
DD)<>SW THEN 3430
3420 ZZ=ZZ+1
3430 IF B(AA)+F(BB)+J(CC)+N(
DD)<>SW THEN 3450
3440 ZZ=ZZ+1
3450 IF C(AA)+G(BB)+K(CC)+O(
DD)<>SW THEN 3470
3460 ZZ=ZZ+1
3470 IF D(AA)+H(BB)+L(CC)+P(
DD)<>SW THEN 3490
3480 ZZ=ZZ+1
3490 IF A(AA)+F(BB)+K(CC)+P(
DD)<>SW THEN 3510
3500 ZZ=ZZ+1
3510 IF D(AA)+G(BB)+J(CC)+M(
DD)<>SW THEN 3530
3520 ZZ=ZZ+1
3530 NEXT X
3540 IF A(1)+A(2)+A(3)+A(4)<
>SW THEN 3560
3550 ZZ=ZZ+1
3560 IF B(1)+B(2)+B(3)+B(4)<
>SW THEN 3580
3570 ZZ=ZZ+1
3580 IF C(1)+C(2)+C(3)+C(4)<
>SW THEN 3600
3590 ZZ=ZZ+1
3600 IF D(1)+D(2)+D(3)+D(4)<
>SW THEN 3620
3610 ZZ=ZZ+1
3620 IF E(1)+E(2)+E(3)+E(4)<
>SW THEN 3640
3630 ZZ=ZZ+1
3640 IF F(1)+F(2)+F(3)+F(4)<
>SW THEN 3660
3650 ZZ=ZZ+1
3660 IF G(1)+G(2)+G(3)+G(4)<
>SW THEN 3680
3670 ZZ=ZZ+1
3680 IF H(1)+H(2)+H(3)+H(4)<
>SW THEN 3700
3690 ZZ=ZZ+1
3700 IF J(1)+J(2)+J(3)+J(4)<
>SW THEN 3720
3710 ZZ=ZZ+1
3720 IF K(1)+K(2)+K(3)+K(4)<
>SW THEN 3740
3730 ZZ=ZZ+1
3740 IF L(1)+L(2)+L(3)+L(4)<
>SW THEN 3760
3750 ZZ=ZZ+1
3760 IF M(1)+M(2)+M(3)+M(4)<
>SW THEN 3780
3770 ZZ=ZZ+1
3780 IF N(1)+N(2)+N(3)+N(4)<
>SW THEN 3800
3790 ZZ=ZZ+1
3800 IF O(1)+O(2)+O(3)+O(4)<
>SW THEN 3820
3810 ZZ=ZZ+1
3820 IF P(1)+P(2)+P(3)+P(4)<
>SW THEN 3840
3830 ZZ=ZZ+1
3840 ZV=ZV+1
3850 ON ZV GOTO 3860,3910
3860 ZD=ZZ
3870 ZZ=0
3880 SW=600
3890 RESTORE 4640
3900 GOTO 3310
3910 G$=H$
3920 Z=2
3930 S=10
3940 GOSUB 3080
3950 CALL HCHAR(2,5,140)
3960 CALL HCHAR(2,19,150)
3970 IF ZD<20 THEN 4000
3980 CALL HCHAR(2,7,50)
3990 ZD=ZD-20
4000 IF ZD<10 THEN 4030
4010 CALL HCHAR(2,7,49)
4020 ZD=ZD-10
4030 ZD=ZD+48
4040 CALL HCHAR(2,8,ZD)
4050 IF ZZ<20 THEN 4080
4060 CALL HCHAR(2,21,50)
4070 ZZ=ZZ-20
4080 IF ZZ<10 THEN 4110
4090 CALL HCHAR(2,21,49)
4100 ZZ=ZZ-10
4110 ZZ=ZZ+48
4120 CALL HCHAR(2,22,ZZ)
4130 ZZ=0
4140 ZD=0
4150 ZA=0
4160 ZV=0
4170 RESTORE 4640
4180 RETURN
4190 REM BERECHNUNG FUER DEN
'TI-99/4A' ZUR SETZUNG DER
 SPIELSTEINE
4210 IF A(1)+A(2)<>270 THEN
4230
4220 GOTO 1200
4230 IF B(1)+B(2)<>270 THEN
4250
4240 GOTO 1300
4250 IF C(1)+C(2)<>270 THEN
4270
4260 GOTO 1410
4270 IF D(1)+D(2)<>270 THEN
4290
4280 GOTO 1500
4290 IF E(1)+E(2)<>270 THEN
4310
```

```
4300 GOTO 1590
4310 IF F(1)+F(2)<>270 THEN
4330
4320 GOTO 1680
4330 IF G(1)+G(2)<>270 THEN
4350
4340 GOTO 1770
4350 IF H(1)+H(2)<>270 THEN
4370
4360 GOTO 1860
4370 IF I(1)+I(2)<>270 THEN
4390
4380 GOTO 1950
4390 IF J(1)+J(2)<>270 THEN
4410
4400 GOTO 2040
4410 IF K(1)+K(2)<>270 THEN
4430
4420 GOTO 2130
4430 IF L(1)+L(2)<>270 THEN
4450
4440 GOTO 2220
4450 IF M(1)+M(2)<>270 THEN
4470
4460 GOTO 2310
4470 IF N(1)+N(2)<>270 THEN
4490
4480 GOTO 2400
4490 IF O(1)+O(2)<>270 THEN
4510
4500 GOTO 2490
4510 IF P(1)+P(2)<>270 THEN
4530
4520 GOTO 2580
4530 RANDOMIZE
4540 KK=(RND*20)-2
4550 IF KK<1 THEN 4540
4560 IF KK>16 THEN 4540
4570 ON KK GOTO 1200,1300,14
10,1500,1590,1680,1770,1860,
1950,2040,2130,2220,2310,240
0,2490,2580
4580 REM  DATEN FUER DEN
4590 REM BILDAUFBAU
4600 DATA 17,1,15,3,13,5,11,
7
4610 DATA 13,3,11,5,9,7,7,9,
13,10,11,12,9,14,7,16,13,17,
11,19,9,21,7,23,13,24,11,26,
9,28,7,30
4620 REM DATEN ZUR
4630 REM MUEHLENAUSWERTUNG
4640 DATA 1,1,1,1,2,2,2,2,3,
3,3,3,4,4,4,4,1,2,3,4,4,3,2,
1
4650 REM  ENDE DES LISTINGS
```

```
10 REM*********************
11 REM*                   *
12 REM* DREIECK & KREIS-  *
13 REM*  BERECHNUNGEN     *
14 REM*  Copyright by     *
15 REM* Juergen Bruehne   *
16 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                   *
23 REM  Speicherbelegung *
24 REM*     4279 Bytes    *
25 REM *********************
26 REM
190 CALL CLEAR
200 PRINT " 1-> DREIECKBEREC
HNUNG (^)": : :" 2-> KREISBE
RECHNUNG": : :" 3-> PROGRAMM
ENDE": : : :
210 CALL KEY(0,K,S)
220 IF S=0 THEN 210
230 CALL CLEAR
240 IF K=49 THEN 280
250 IF K=50 THEN 1190
260 IF K>51 THEN 210
270 END
280 PRINT " A=ANKATHETE": :"
 C=GEGENKATHETE": :"H=HYPOT
HENUSE": :" a=ALPHA": :" b=B
ETA": :" g=GAMMA (/\=90')"
290 PRINT : :".........g....
......................./ \...
.....................A...G"
300 PRINT "....../.....\....
.............a---H---b"
310 PRINT : :"WEITER MIT 'EN
TER'":"RUECKSPRUNG MIT 'R'"
320 CALL KEY(0,K,S)
330 IF S=0 THEN 320
340 IF K=82 THEN 100
350 CALL CLEAR
360 PRINT : :" BITTE WAEHLEN
 SIE !";" ----------------"
: : : :" [1] HYPOTHENUSE + A
NKATHETE": :" [2] HYPOTHENUS
E + GEGENKA.": :
370 PRINT " [3] HYPOTHENUSE
+ ALPHA": :" [4] HYPOTHENUSE
 + BETA": :" [5] ANKATHETE +
 ALPHA": :" [6] ANKATHETE +
BETA": :
380 PRINT " [7] GEGENKATHETE
 + ALPHA": :" [8] GEGENKATHE
TE + BETA": :" [9] ANKATHETE
 + GEGENKA."
390 CALL KEY(0,K,S)
400 IF S=0 THEN 390
410 V=90
420 IF K<49 THEN 390
430 IF K>57 THEN 390
440 CALL CLEAR
```

```
450 ON K-48 GOTO 460,520,570
,640,710,780,850,920,990
460 INPUT "HYPOTHENUSE ?":HY
470 INPUT "ANKATHETE ?":AK
480 IF AK>=HY THEN 470
490 E=HY^2-AK^2
500 GK=E^(1/2)
510 GOTO 1100
520 INPUT "HYPOTHENUSE ?":HY
530 INPUT "GEGENKATHETE ?":G
K
540 IF GK>=HY THEN 530
550 AK=(HY^2-GK^2)^(1/2)
560 GOTO 1100
570 INPUT "HYPOTHENUSE ?":HY
580 INPUT "ALPHA ?":AL
590 IF AL>=V THEN 580
600 CA=COS(AL*(4*ATN(1))/180
)
610 AK=HY*CA
620 GK=(HY^2-AK^2)^(1/2)
630 GOTO 1100
640 INPUT "HYPOTHENUSE ?":HY
650 INPUT "BETA ?":BE
660 IF BE>=V THEN 650
670 CB=COS(BE*(4*ATN(1))/180
)
680 GK=CB*HY
690 AK=(HY^2-GK^2)^(1/2)
700 GOTO 1100
710 INPUT "ANKATHETE ?":AK
720 INPUT "ALPHA ?":AL
730 IF AL>=V THEN 720
740 CA=COS(AL*(4*ATN(1))/180
)
750 HY=AK/CA
760 GK=(HY^2-AK^2)^(1/2)
770 GOTO 1100
780 INPUT "ANKATHETE ?":AK
790 INPUT "BETA ?":BE
800 IF BE>=V THEN 790
810 SB=SIN(BE*(4*ATN(1))/180
)
820 HY=AK/SB
830 GK=(HY^2-AK^2)^(1/2)
840 GOTO 1100
850 INPUT "GEGENKATHETE ?":G
K
860 INPUT "ALPHA ?":AL
870 IF AL>=V THEN 860
880 SA=SIN(AL*(4*ATN(1))/180
)
890 HY=GK/SA
900 AK=(HY^2-GK^2)^(1/2)
910 GOTO 1100
920 INPUT "GEGENKATHETE ?":G
K
930 INPUT "BETA ?":BE
940 IF BE>=V THEN 930
950 CB=COS(BE*(4*ATN(1))/180
)
960 HY=GK/CB
970 AK=(HY^2-GK^2)^(1/2)
980 GOTO 1100
990 INPUT "ANKATHETE ?":AK
1000 INPUT "GEGENKATHETE ?":
GK
1010 HY=(AK^2+GK^2)^(1/2)
1020 GOTO 1100
1030 PRINT "HYPOTHENUSE",HY:
 :"ANKATHETE",AK: :"GEGENKAT
HETE",GK: :"ALPHA",AL: :"BET
A",BE: :"GAMMA",90: :"COS a"
,CA: :
1040 PRINT "SIN a",SA: :"TAN
 a",TA: :"COS b",CB: :"SIN b
",SB: :"TAN B",TB
1050 CALL KEY(0,K,S)
1060 IF S=0 THEN 1050
1070 IF K<>82 THEN 360
1080 CALL CLEAR
1090 GOTO 190
1100 CA=AK/HY
1110 SA=GK/HY
1120 CB=SA
1130 SB=CA
1140 TA=SA/CA
1150 TB=SB/CB
1160 AL=ATN(TA)*(180/(4*ATN(
1)))
1170 BE=ATN(TB)*(180/(4*ATN(
1)))
1180 GOTO 1030
1190 PI=(4*ATN(1))
1200 PRINT : : :"WELCHER WER
T IST GEGEBEN ?": :" [1] DUR
CHMESSER(d)": :" [2] RADIUS(
r)": :" [3] UMFANG(u)": :
1210 PRINT " [4] FLAECHENINH
ALT (A)": :" [5] FORMELN": :
 :"FUER pi=";PI: :" [6] WENN
 FUR pi EINE ANDERE": :
1220 PRINT " ZAHL EINGESETZT
 WERDEN SOLL": :" [7] RUECKS
PRUNG": :
1230 CALL KEY(0,K,S)
1240 IF S=0 THEN 1230
1250 IF K<49 THEN 1230
1260 IF K>55 THEN 1230
1270 CALL CLEAR
1280 PRINT "BITTE ZAHL EINGE
BEN": :"DENKEN SIE DARAN FUE
R EIN": :" ',' EIN '.' ZU SE
TZEN": : : :
1290 ON K-48 GOTO 1320,1370,
1420,1470,1520,1300,100
1300 INPUT "fuer pi=":PI
1310 GOTO 1200
1320 INPUT "DURCHMESSER (d):
":D
1330 R=D/2
1340 U=D*PI
```

```
1350 A=D^2*PI/4
1360 GOTO 1580
1370 INPUT "RADIUS (r):":R
1380 D=2*R
1390 U=D*PI
1400 A=R^2*PI
1410 GOTO 1580
1420 INPUT "UMFANG (u):":U
1430 D=U/PI
1440 R=D/2
1450 A=R^2*PI
1460 GOTO 1580
1470 INPUT "FLAECHENINHALT (
A):":A
1480 R=(A/PI)^(1/2)
1490 D=R*2
1500 U=D*PI
1510 GOTO 1580
1520 CALL CLEAR
1530 PRINT ".....FORMELN
:":".....---------": : :"
d=DURCHMESSER": :"r=RADIUS":
 :"u=UMFANG": :"A=FLAECHENIN
HALT"
1540 PRINT : :" d=2*r","r=d/
2": :" u=d*pi","u=2*r*pi": :
" A=r^2*pi","A=d^2*pi/4": :
:"WEITER MIT 'ENTER'": :
1550 CALL KEY(0,K,S)
1560 IF K<>13 THEN 1550
1570 GOTO 1190
1580 CALL CLEAR
1590 PRINT "DURCHMESSER :":"
  d:",D: :"RADIUS :":"  r:",
R: :"UMFANG :":"  u:",U: :"F
LAECHENINHALT :":"  A:",A: :
"WOLLEN SIE EINEN"
1600 PRINT "KREISABSCHNITT B
ERECHNEN ?": :" (J/N) ?":
1610 CALL KEY(0,K,S)
1620 IF S=0 THEN 1610
1630 IF K<>74 THEN 1200
1640 PRINT "BITTE GEBEN SIE
DEN": :"WINKEL EIN !": : :
1650 INPUT "WINKEL : ":W
1660 IF W>360 THEN 1650
1670 U=U*W/360
1680 A=A*W/360
1690 PRINT : : : :"WINKEL :"
,W: :"FLAECHENINHALT :": :"
 Aw:",A: :"UMFANG :": :"  uw
:",U: : :"WEITER MIT 'ENTER'
 ": :
1700 CALL KEY(0,K,S)
1710 IF K<>13 THEN 1700
1720 GOTO 1200
```


ACHTUNG!
089/1298011 Ab 1.10.85!

BÖRSE

Achtung ! Ich drucke kostenlos Eure Listings. Rückporto! Programme an Mcihael Wiese, Postfach 3343, 4950 Minden 2

Verkaufe Programme in TI und XBasic. Liste bei Hans Boyer Jr., Eichenstr. 17, 8075 Vohburg gegen RP.

Verk. TI-99 + Ext. Basic + Rec-Kabel + Joyst. + Module: Buchungsjournal, Munch Man, TI Invaders, Chissh. Trail, Addition, Subtr., Multipl. + Basickurs + viel Softw. auf Kass. + viele Bücher v. Zeitschr. Auch einzeln! VB tel. 09171/1716

Suche TI-User, möglichst mit Assemblerkenntnissen im Raum Salzgitter zum Prg- und Moduletausch
Oliver Moldenhauer, Humboldtallee 19, 3320 SZ-1, Tel.: 05341/54411

---

* Suche Adventuremodul * mit Cassetten!
* Tausche Programme * nur Ex.-Basic!
* Tel.: 02103/47208 *

Module f. TI99/4A ges. Angebote an A. Froberg Klopstockstr. 4, 2850 Bremerhaven. P.S. Zahle per Nachnahme

---

Verkaufe/Tausche TI-org.-Laufw. 250. Inv., Parsec je 30,— Suche Speechy-Sy. bis 80,— Tel.: 07240/5300

Verkaufe 3 Original TI-Revue Programmcassetten aus TI-Spezial Paket 4 Hefte 6+7 DM 20,— 02271/65673

Suche für 99/4A preiswert Minimem, 32K Erweiterung extern, J. Rybarski, 02271/65673

Achtung: Habe vor TI-Club zwecks Progr.-Austausch zu gründen. M. Nagler, Tradtstr. 14, 8492 Fürth, 09973/1517

Verkaufe mein Grafik-Adventure „Osus-Shan 1"; Test in Adv-Magazin: Grafik „Sehr gut" deutschsprachig! In Ti-Bsic! Auch andere Spiele und Anwenderprogramme in Gratis. Info bei: W. Mskens, Kranenburgerstr. 224/4180 Goch 6

TI 99/4A+rec+Kabel+Parsec+TI Invader 350 DM, Ex Basic 150 DM, Speechsyn 120 DM, TI Writer 220 DM, TE II 50 DM alle Preise VHG, Rolf Dönselmann Skagerrakstr. 10c, 2960 Aurich Tel. 04941/4101 App. 278

TI 99/4A + Selbstbauer W-Box / Centronics, 32K, Epromprogrammer, GP100, viel Literatur + Programme. 02721/5395

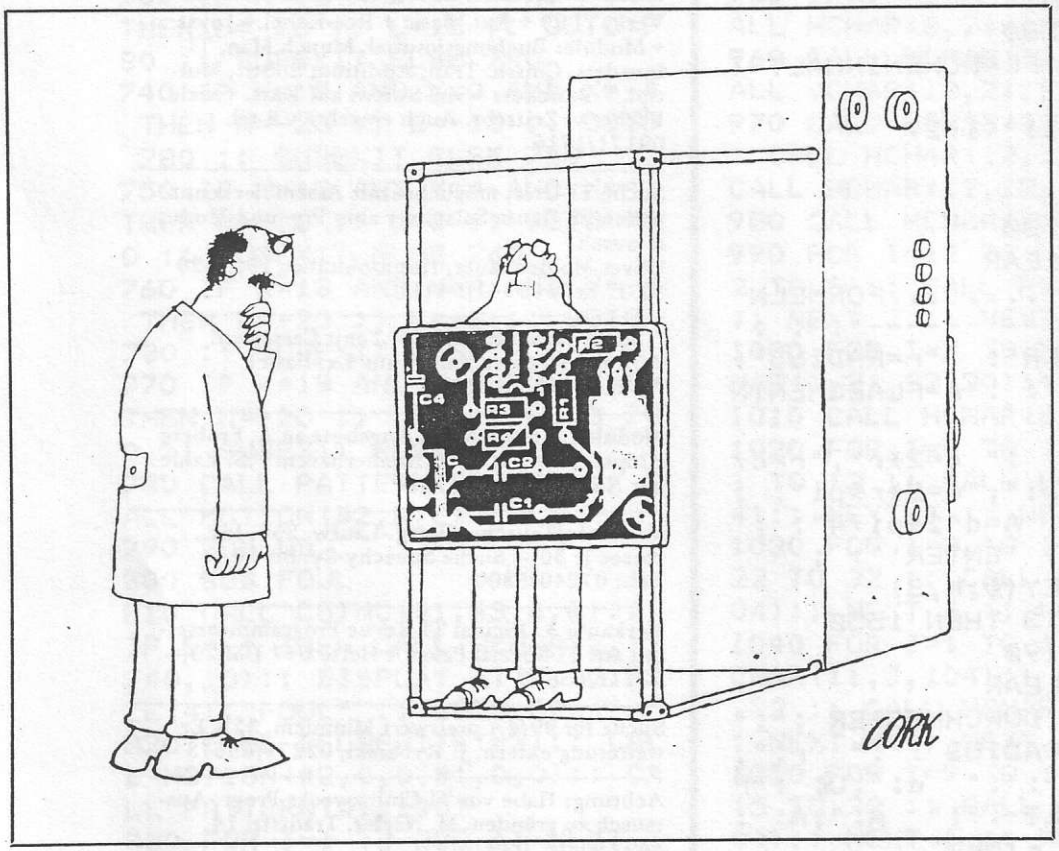
Verk. TI99/4A+Ext. Basic+Rec. Kabel + TI-Joyst. + Handbuch + Data Becker Buch / viele List. u.Progr. Cass. + 8 TI-Revue für DM 400,— Nur komplett. B. Schmitt, Tel. 09721/33843 nach 18 Uhr

---

Suche Amateurfunk-SW jegl. Art: Außerdem: Allg. Literatur zur Assemblerprogrammierung (Bücher/Tips & Tricks)
Hartleib, Hardenbergstr. 16, 3500 Kassel

Exbasic 2 + Painter + Hardcopy, div. Bücher sowie Kass. Kabel günstig !!!
Tel. 06638/1503

BOOGABOO DER MUSIKALISCHE FLOH



Boogaboo ist ein kleiner Floh, und dazu noch ein sehr musikalischer. Deshalb nimmt er alle Gefahren auf sich und sammelt seine verlorengegangenen, verstreuten Noten auf. Da er sich aber in einem vierstöckigen Höhlengewölbe befindet, ist dies nicht allzu leicht. Immerfort drohen fallende Felsbrocken ihn zu erschlagen oder fliegende Notenschlüssel ihn aufzuspießen. Zur Bewegung und zu seiner Hilfe stehen ihm aber Aufzüge und Gleitböden zur Verfügung, auf denen er während des Fahrens völlig sicher ist.

Nach dem Erscheinen des Titelbilds wird gefragt, ob eine Spielerläuterung gewünscht wird. Ist dies nicht der Fall, beginnt das Spiel. Der Spieler steuert Boogaboo mit dem Joystick 2. Durch Drücken des Hebel nach oben oder unten wird der Aufzug benutzt. Er verbindet — bis auf einen Fall im 3. Bild — immer nur 2 Etagen miteinander. Gleichzeitiges Drücken nach rechts oder links und Betätigen des Aktionsknopfes am Joystick bewirkt das Fahren mit dem Gleitboden. Aufgabe ist es, alle verstreuten 18 Noten aufzusammeln, wobei jedoch immer nur drei auf dem Bildschirm erscheinen. Dabei ist sowohl auf die schwindende Zeit als auch auf zahlreiche Hindernisse zu achten. Zum einen muß man aufpassen, daß Boogaboo immer festen Boden unter den Füßen hat, da er ansonsten unweigerlich abstürzt. Zum andern werden ihm fallende Mauerteile und waagerechte Notenschlüssel gefährlich. Lediglich ein aufrechter Notenschlüssel, der jedoch nur zweimal in einem Bild erscheint, bringt Bonuspunkte in Höhe der Hälfte der gegenwärtigen Punktzahl. Fängt Boogaboo diesen Notenschlüssel, so ertönt nach Sammeln aller Noten eine Melodie und die restliche Zeit wird zur Punktzahl addiert. Fängt er ihn nicht, gibt es keinen Bonus und die Melodie kann aufgrund des fehlenden Notenschlüsses nicht angestimmt werden.

Sind alle 18 Noten gesammelt, erscheint ein schwierigeres Bild und Boogaboo beginnt seine Sammelaktion von Neuem. Das Spiel ist beendet, wenn alle 4 Leben unachtsam verspielt worden sind.

Boogaboo ist auf dem TI-99/4A mit EXTENDED Basic programmiert. Ein Großteil der verwendeten Routinen wurde in Unterprogrammen (Subprogramms) erstellt. Dadurch wird neben der Übersichtlichkeit in einigen Fällen auch die Ausführungsgeschwindigkeit erhöht.

Bei der Programmerstellung habe ich bewußt auf effektvolle Graphikhascherei verzichtet. So stellt das vorliegende Programm in Geschwindigkeit, Graphik, Sound und Abwechslung einen — wie ich meine — den Fähigkeiten des TI entsprechenden optimalen Kompromiß dar. In dieser Weise werden denn auch nach Initialisierung ca. 13 600 Bytes belegt, so daß im Interesse eines reibungslosen Programmablaufes davon abzusehen ist, eventuelle größere Veränderungen vorzunehmen. Dementsprechend waren bei mir während des Endstadiums der Entwicklung Programmverluste und Systemabstürze an der Tagesordnung.

Wer dennoch Boogaboo eine größere Chance geben bzw. seine Geschwindigkeit erhöhen will, kann dies durch Ändern der Zeile 600 bewirken. Hier muß nur in den

Bitte lesen Sie weiter auf Seite 123

```
   **********************
10 ! *                    *
11 ! *      BOOGABOO      *
12 ! *                    *
13 ! *    Copyright by    *
14 ! *   Peter Krawinkel  *
15 ! *                    *
16 ! * Benoetigte Geraete *
19 ! *  TI99/4A  Konsole  *
20 ! *     Ext. Basic     *
21 ! *    Joystick (1)    *
22 ! *                    *
23 ! *  Speicherbelegung  *
26 ! *    11065 Bytes     *
27 ! *                    *
28 ! **********************
29 !
30 !
220 CALL CLEAR :: CALL SCREE
N(5):: CALL MAGNIFY(3):: RAN
DOMIZE
230 FOR I=1 TO 12 :: CALL CO
LOR(I,16,1):: NEXT I :: CALL
 COLOR(10,16,7)
240 OPTION BASE 1 :: DIM P(3
,6,2),ZE(18),SP(18)
250 !
260 ! CHAR-DEFINITIONEN
270 !
280 CALL CHAR(120,"FF7F3F3F1
20A080D32743012100807 00FFFEF
CF8489860920C0868281010 20C")
290 CALL CHAR(124,"0040C0201
0080402030 50F057C2310380004 0
6081020408080 40E0407C881038"
)
300 CALL CHAR(128,"000000001
028C44203050F053C43808000000
0001028468480 40E04078840202"
)
310 CALL CHAR(132,"1C08113E0
2070201010204081060200 01C08C
43EA0F0A0C040201008040302")
320 CALL CHAR(36,"0102020101
0101070911110907010106C02020
20408000C020101020C")
330 CALL CHAR(136,"000000060
990907F10100906"&RPT$("0",18
)&"8086FAA2920E")
340 CALL CHAR(140,RPT$("0",1
6)&"FFFF7F3F1F0F"&RPT$("0",2
0)&"FFFFFEFCF8F")
350 CALL CHAR(104,"FF202020F
F020202",112,"FF7F3F3F",113,
"FFFEFCF8",114,"")
360 CALL CHAR(115,"070404040
41C3C18",116,"1F1111111177FF
66",117,"07040404041C2418",1
18,"1F11111111779966",119,"C
34224183C5A42")
370 GOTO 620
380 PU,R,R1=0 :: MA=3
390 !
400 ! RUNDENBEGINN
410 !
420 R=R+1 :: R1=R1+1 :: IF R
>3 THEN R=1
430 AN,NS,A,B,BN,BC=0 :: TI=
250
440 CALL LIFT(R,P(,,),AW()):
: CALL NOTVER(ZE(),SP(),R)::
 GOSUB 1140
450 CALL BILD(R):: CALL SPRI
TE(#1,124,16,129,57):: CALL
COLOR(#2,2,#3,2,#4,2,#5,2,#6
,11,#7,11)
460 !
470 ! HAUPTPROGRAMM
480 !
490 CALL TREFF(A,B,BN):: CAL
L COINC(#1,#10,10,CU):: IF C
U THEN 820 ELSE GOSUB 570
500 CALL FNS(A,B,R1,SX,BN,BC
):: CALL COINC(#1,#10,10,CU)
:: IF CU THEN 820 ELSE GOSUB
 570
510 CALL TREFF(A,B,BN):: CAL
L COINC(#1,#10,10,CU):: IF C
U THEN 820 ELSE GOSUB 570
520 CALL FALL(A,B,R1,SY):: C
ALL COINC(#1,#10,10,CU):: IF
 CU THEN 820 ELSE GOSUB 570
530 TI=TI-1 :: IF TI<=0 THEN
 820 ELSE GOTO 490
540 !
550 ! BEWEGUNG
560 !
570 CALL KEY(2,K,S):: CALL J
OYST(2,X,Y):: IF Y<>0 OR S*X
 THEN CALL MOV(R,P(,,),AW(),
Y,X)
580 CALL POSITION(#1,SX,SY):
: CALL GCHAR((SX+7)/8,(SY+11
)/8,P1):: CALL GCHAR((SX+23)
/8,(SY+11)/8,P2)
590 IF P1<>32 THEN GOSUB 121
0 ELSE IF P2=32 THEN 820
600 CALL PATTERN(#1,128):: C
ALL LOCATE(#1,SX,SY+2*X):: C
ALL PATTERN(#1,124):: CALL L
OCATE(#1,SX,SY+2*X)
610 RETURN
620 !
630 ! TITELBILD
640 !
650 CALL CHAR(96,"00000F1F3F
3F3C3C",97,RPT$("3C",8),98,"
0000FFFFFFFF",99,"3C3C3F3F1F
0F",64,"3C4299A1A199423C")
660 CALL CHAR(100,"0000F0F8F
CFC3C3C",101,"3C3CFCFCF8F0",
102,"3C78F0F0F0F0783C",103,"
3C3C3C3C3C3C")
```

```
670 CALL SCREEN(5):: CALL CO
LOR(9,16,1)
680 PRINT "..'bd'bd'bd'b 'bd
'bd'bd'bd"
690 PRINT "..a aa aa aa  a a
a aa aa a"
700 PRINT "..abfa aa aabdaba
abfa aa a"
710 PRINT "..a aa aa aa aa a
a aa aa a"
720 PRINT "..cbecbecbecbeg g
cbecbecbe": : : :"..@ BY P.K
RAWINKEL APRIL'85": : : :
730 PRINT ".....hh.....
.hh": : : :".....hh.....
..hh": : : : : : :
740 DISPLAY AT(20,3)BEEP:"PR
ESS..." :: DISPLAY AT(22,3)B
EEP:"1 > FOR INSTRUCTIONS" :
: DISPLAY AT(24,3)BEEP:"2 >
TO PLAY "
750 CALL SPRITE(#2,140,2,121
,65,#7,140,11,89,97,#3,140,2
,89,177,#6,140,11,121,145,#1
,124,16,113,153)
760 CALL MUSIC(58)
770 CALL KEY(0,K,S):: IF S=0
 THEN 770 ELSE IF K=49 THEN
1370
780 IF K=50 THEN CALL CLEAR
:: CALL DELSPRITE(ALL):: GOT
O 380 ELSE 770
790 !
800 ! ABSTURZ
810 !
820 CALL DELSPRITE(#10):: IF
 BN=1 THEN 950 ELSE IF TI>0
THEN 840
830 DISPLAY AT(8,4):"DIE ZEI
T WAR BEGRENZT !" :: DISPLAY
 AT(12,4):"WOHL GESCHLAFEN,
ODER ?" :: TI=250
840 FOR I=1 TO 4 :: CALL SOU
ND(-139,110,5,111,5,112,5)::
 CALL PATTERN(#1,128)
850 CALL SOUND(-139,440,5,44
1,5,442,5):: CALL PATTERN(#1
,124):: NEXT I
860 CALL PATTERN(#1,132):: C
ALL MOTION(#1,9,0):: FOR I=6
60 TO 110 STEP -10 :: CALL P
OSITION(#1,SX,SY)
870 CALL SOUND(-139,I,5,I+1,
5,I+2,5):: IF SX>192 THEN 89
0
880 NEXT I
890 CALL DELSPRITE(#1):: FOR
 I=1 TO 4 :: CALL SOUND(169,
110,5,220,5):: NEXT I
900 MA=MA-1 :: IF MA<0 THEN
1290 ELSE A,B=0 :: CALL SCOR
E(PU,HI,MA,TI)
910 CALL HCHAR(8,3,32,26)::
CALL HCHAR(12,3,32,26):: GOT
O 450
920 !
930 ! BONUS
940 !
950 FOR I=1 TO 3 :: FOR J=22
0 TO 330 STEP 10 :: CALL SOU
ND(-139,J,5,J+1,5,J+2,5):: N
EXT J :: NEXT I
960 CALL SPRITE(#11,36,16,17
8,9)
970 FOR I=PU TO PU*1.5 STEP
10*R1 :: DISPLAY AT(22,1)SIZ
E(11):USING "SCORE #####":I
:: CALL SOUND(39,111,5):: NE
XT I
980 BN,A,B=0 :: BC=2 :: PU=P
U*1.5 :: CALL SCORE(PU,HI,MA
,TI):: GOTO 490
990 !
1000 ! NAECHSTE RUNDE
1010 !
1020 CALL DELSPRITE(#10):: C
ALL POSITION(#1,SX,SY)
1030 FOR I=1 TO 6 :: CALL PA
TTERN(#1,128):: CALL LOCATE(
#1,SX-6,SY):: CALL SOUND(-16
9,440,5,444,5)
1040 FOR J=1 TO 35 :: NEXT J
 :: CALL LOCATE(#1,SX,SY)::
CALL PATTERN(#1,124)
1050 CALL SOUND(-169,220,5,2
22,5):: FOR J=1 TO 35 :: NEX
T J :: NEXT I
1060 CALL POSITION(#11,SX,SY
):: IF SX<>178 THEN 1090 ELS
E IF R1/3=INT(R1/3)THEN CALL
 MUSIC(58)ELSE CALL MUSIC(30
)
1070 FOR I=PU TO PU+TI STEP
10 :: DISPLAY AT(22,7)SIZE(5
):USING "#####":I
1080 DISPLAY AT(24,26)BEEP:U
SING "###":PU+TI-I :: NEXT I
 :: PU=PU+TI
1090 TI=0 :: CALL SCORE(PU,H
I,MA,TI):: CALL DELSPRITE(AL
L)
1100 CALL HCHAR(6,1,32,480):
: CALL HCHAR(24,4,32,18):: G
OTO 420
1110 !
1120 ! NOTEN SETZEN
1130 !
1140 NS=NS+3 :: IF NS>=21 TH
EN 1020
1150 FOR I=0 TO 2 :: CALL HC
HAR(ZE(NS-I)*4+1,SP(NS-I),IN
T(RND*4)+115)
```

```
1160 CALL SOUND(-99,220,5,22
1,5,222,5):: NEXT I
1170 TI=TI-3 :: CALL SCORE(P
U,HI,MA,TI):: RETURN
1180 !
1190 ! NOTEN SAMMELN
1200 !
1210 CALL HCHAR((SX+7)/8,(SY
+11)/8,32):: CALL SOUND(-99,
110,5,333,5)
1220 CALL HCHAR(24,NS+AN+1,I
NT(RND*4)+115):: PU=PU+R1*10
 :: TI=TI-1
1230 CALL SCORE(PU,HI,MA,TI)
1240 AN=AN+1 :: IF AN=3 THEN
 AN=0 :: GOSUB 1140
1250 RETURN
1260 !
1270 ! END OF GAME
1280 !
1290 CALL DELSPRITE(ALL):: C
ALL CLEAR :: CALL SCORE(PU,H
I,MA+1,TI):: DISPLAY AT(2,1)
:">>>>>>G A M E  O V E R<<<<
<<"
1300 DISPLAY AT(7,1):"VIER F
LOEHE VERHEIZT !": :"WOHL BL
IND WAS ?"
1310 IF PU>HI THEN CALL SCOR
E(PU,PU,MA+1,TI):: HI=PU ELS
E 1340
1320 DISPLAY AT(12,1):"TROTZ
DEM....................." ::
DISPLAY AT(14,1):"GRATULATIO
N.....HIGHSCORE !!"
1330 CALL MUSIC(58)
1340 DISPLAY AT(17,1):"NEUES
 SPIEL...NEUES GLUECK !": :"
BITTE >J< ODER >N< DRUECKEN!
"
1350 CALL KEY(0,K,S):: IF S=
0 THEN 1350
1360 CALL CLEAR :: IF K=78 O
R K=110 THEN STOP ELSE 380
1370 !
1380 ! SPIELERLAEUTERUNG
1390 !
1400 CALL DELSPRITE(ALL):: C
ALL CLEAR
1410 PRINT "BOOGABOO, DER MU
SIKALISCHE": :"FLOH, WILL SE
INE VERLORENEN": :"NOTEN WIE
DER AUFSAMMELN.": : : :
1420 PRINT "HELFEN SIE IHM D
ABEI, INDEM": :"SIE DIE AUFZ
UEGE (JOYSTICK": :"NACH OBEN
 / UNTEN) ODER": :
1430 PRINT "DIE GLEITBOEDEN
(JOYSTICK": :"RECHTS / LINKS
 + FIRETASTE )": :"BENUTZEN.
": : :
1440 CALL KEY(0,K,S):: IF S=
0 THEN 1440
1450 PRINT "LASSEN SIE SICH
JEDOCH NICHT": :"VON FALLEND
EN MAUERTEILEN": :"ODER LIEG
ENDEN NOTENSCHLUES-": :
1460 PRINT "SELN ERWISCHEN !
": :"NUR AUFRECHTE NOTENSCHL
UES-": :"SEL BRINGEN IHNEN D
IE HAELF-": :
1470 PRINT "TE IHRER PUNKTZA
HL ALS BONUS": :"UND SPAETER
 DIE RESTLICHE": :"ZEIT IN P
UNKTEN.": :
1480 PRINT "DIESER NOTENSCHL
UESSEL ER-": :"SCHEINT IM EI
NER RUNDE NUR": :"2 MAL."
1490 CALL KEY(0,K,S):: IF S=
0 THEN 1490
1500 PRINT : :"OHNE DEN NOTE
NSCHLUESSEL ": :"GIBT'S >KEI
NEN< BONUS, AUCH": :"NICHT A
M RUNDENENDE !": : :
1510 PRINT "EIN TIP !": :"VO
LLKOMMEN SICHER SIND SIE ":
:"NUR, WAEHREND SIE AUFZUG":
 :"ODER GLEITBODEN FAHREN !"
: : : : :
1520 PRINT ">ALPHA-LOCK< LOE
SEN !": :"PRESS ANY KEY TO B
EGIN....."
1530 CALL KEY(0,K,S):: IF S=
0 THEN 1530 ELSE CALL CLEAR
:: GOTO 380
1540 !
1550 ! DATEN AUFZUEGE
1560 !
1570 DATA 137,105,32,105,73,
32,73,41,32,73,41,32,81,113,
32,137,153,16
1580 DATA 137,105,32,105,73,
32,73,41,32,73,41,32,57,81,2
4,81,145,64
1590 DATA 137,105,32,105,73,
32,105,41,64,73,41,32,113,16
1,48,97,161,64
1600 !
1610 ! DATEN DISPLAY>1,2,3<
1620 !
1630 DATA "hhhhhhhhhh..hhhhh
hhhhhrrhhhh",7
1640 DATA "hhh..hhhhhhrrhhhrr
..hhh..hhhh",11
1650 DATA "hhhrrhhhhhhhhhhhh
hhh..hhhhhh",15
1660 DATA "hhhhhhhhrr....hhh
hhhrrhhhhhh",19
1670 DATA "hhhhhhhh.....hhh
hhhhhhhhhh",20
1680 DATA 137,177,105,41,73,
97,41,193,137,81,73,137
```

```
1690 DATA "hhh..hhhrr.....
.hhhhrrhhhh",7
1700 DATA "hhhrrhhhhhhhhhhhh
h..hh..hhhh",11
1710 DATA "hhhhh...rrhhh..hh
hrrhhhhhhhh",15
1720 DATA "hhhhhhhhhhhhhrrhh
hhhhhhhhhhh",19
1730 DATA "hhhhhhhhhhhhh..hh
hhhhhhhhhhh",20
1740 DATA 137,121,105,161,73
,41,41,193,105,81,41,81
1750 DATA "hhhhhrrhhhhhhhhhh
hhhhhhhhh..",7
1760 DATA "hhhhh..hhhrr.....
...hh..h",11
1770 DATA "hhhhhhhh..hhrr...
...hhrrhhrr",15
1780 DATA "hhhhhhhhrrhhhhhhh
hhhhhhhhhh",19
1790 DATA "hhhhhhhh..hhhhhhh
hhhhhhhhhh",20
1800 DATA 137,81,105,193,105
,225,41,57,105,113,73,97
1810 !
1820 ! DATEN MUSIKNOTEN
1830 !
1840 DATA 2,23,3,10,2,29,1,7
,4,18,3,15,4,29,1,4,2,16,3,1
8,3,4,4,9,1,16,2,23,4,28,3,4
,1,10,1,19
1850 DATA 2,4,4,20,2,19,3,14
,1,22,4,9,1,9,3,5,2,13,4,4,4
,26,1,29,3,6,2,4,1,28,4,24,3
,6,3,27
1860 DATA 1,22,2,27,1,17,3,2
4,4,19,2,10,4,26,3,28,1,4,2,
23,2,4,4,19,3,14,1,23,4,6,2,
27,3,4,3,23
1870 !
1880 ! DATEN MUSIK
1890 !
1900 DATA 5,391,5,523,5,523,
5,587,5,659,5,523,5,659,5,58
7,5,493,5,523,5,523,5,587,5,
659,10,523,5,493
1910 DATA 5,391,5,523,5,523,
5,587,5,659,5,698,5,659,5,58
7,5,523,5,493,5,391,5,440,5,
493,10,523,10,523
1920 DATA 7.5,220,2.5,246,5,
220,5,195,5,220,5,246,5,261,
5,220,7.5,195,2.5,220,5,195,
5,174,10,164,10,195
1930 DATA 7.5,220,2.5,246,5,
220,5,195,5,220,5,246,5,261,
5,220,5,195,5,261,5,246,5,29
3,10,261,5,261
1940 SUB LIFT(R,P(,,),AW())
1950 ON R GOTO 1960,1970,198
0
1960 RESTORE 1570 :: GOTO 19
90
1970 RESTORE 1580 :: GOTO 19
90
1980 RESTORE 1590
1990 FOR I=1 TO 6 :: READ P(
R,I,1),P(R,I,2),AW(I):: NEXT
 I :: SUBEND
2000 SUB BILD(R)
2010 ON R GOTO 2020,2030,204
0
2020 RESTORE 1630 :: GOTO 20
50
2030 RESTORE 1690 :: GOTO 20
50
2040 RESTORE 1750
2050 CALL HCHAR(1,3,104,28):
: DISPLAY AT(2,1):RPT$("xz",
14):: DISPLAY AT(3,1):RPT$("
y{",14)
2060 FOR I=1 TO 5 :: READ A$
,Z :: DISPLAY AT(Z,1):A$ ::
NEXT I
2070 FOR I=2 TO 7 :: READ Z,
S :: CALL SPRITE(#I,140,1,Z,
S):: NEXT I :: SUBEND
2080 SUB MOV(R,P(,,),AW(),Y,
X)
2090 FOR I=2 TO 5 :: CALL CO
INC(#1,#I,8,C(I)):: IF C(I)*
Y THEN 2120
2100 NEXT I
2110 CALL COINC(#1,#6,8,CO1)
:: CALL COINC(#1,#7,8,CO2)::
 IF CO1*X THEN 2150 ELSE IF
CO2*X THEN 2160 ELSE SUBEXIT
2120 CALL POSITION(#I,SX,SY)
:: IF (Y=-4 AND SX=P(R,I-1,1
))OR(Y=4 AND SX=P(R,I-1,2))T
HEN SUBEXIT ELSE 2130
2130 FOR J=SX TO SX-AW(I-1)*
SGN(Y)STEP -Y/2 :: CALL LOCA
TE(#1,J-8,SY,#I,J,SY):: NEXT
 J
2140 CALL HCHAR((SX+15)/8,(S
Y+7)/8,32,2):: CALL HCHAR((S
X+15-SGN(Y)*AW(I-1))/8,(SY+7
)/8,114,2):: SUBEXIT
2150 CALL POSITION(#6,SX,SY)
:: IF (X=4 AND SY=P(R,5,2))O
R(X=-4 AND SY=P(R,5,1))THEN
SUBEXIT ELSE J=5 :: GOTO 217
0
2160 CALL POSITION(#7,SX,SY)
:: IF (X=4 AND SY=P(R,6,2))O
R(X=-4 AND SY=P(R,6,1))THEN
SUBEXIT ELSE J=6
2170 FOR I=SY TO SY+AW(J)*SG
N(X)STEP X/2 :: CALL LOCATE(
#1,SX-8,I,#J+1,SX,I):: NEXT
 I
```

```
2180 CALL HCHAR((SX+15)/8,(S
-Y+7)/8,32,2):: CALL HCHAR((S
X+15)/8,(SY+7+SGN(X)*AW(J))/
8,114,2)
2190 SUBEND
2200 SUB FNS(A,B,C,D,E,F)
2210 IF A/2=INT(A/2)OR B=1 T
HEN SUBEXIT ELSE IF F>1 THEN
 PA=136 :: GOTO 2230
2220 IF RND>.26 THEN PA=136
ELSE PA=36 :: E=1 :: F=F+1
2230 H=8*(-1)^(INT(RND*3)+1)
:: IF H>0 THEN G=17 ELSE G=2
39
2240 CALL SPRITE(#10,PA,16,D
-4,G,0,H+SGN(H)*C):: B=1
2250 SUBEND
2260 SUB TREFF(A,B,C)
2270 CALL POSITION(#10,SX,SY
):: IF SY*SX=0 THEN SUBEXIT
ELSE IF SY<18 OR SY>238 OR S
X>145 THEN CALL DELSPRITE(#1
0)ELSE SUBEXIT
2280 A=A+1 :: B,C=0 :: SUBEN
D
2290 SUB FALL(A,B,C,D)
2300 IF A/2<>INT(A/2)OR B=1
THEN SUBEXIT
2310 X=(INT(D+7)/8)*8-7 :: C
ALL SPRITE(#10,120,16,17,X,9
+C,0):: B=1
2320 SUBEND
2330 SUB NOTVER(ZE(),SP(),A)
2340 ON A GOTO 2350,2360,237
0
2350 RESTORE 1840 :: GOTO 23
80
2360 RESTORE 1850 :: GOTO 23
80
2370 RESTORE 1860
2380 FOR I=1 TO 18 :: READ Z
E(I),SP(I):: NEXT I
2390 SUBEND
2400 SUB SCORE(A,B,C,D)
2410 DISPLAY AT(22,1):USING
"SCORE #####  HI ##### CO ##
#":A,B,RPT$("W",C):: DISPLAY
 AT(24,21):USING "TIME ###":
D
2420 SUBEND
2430 SUB MUSIC(A)
2440 RESTORE 1900
2450 FOR I=1 TO A :: READ B,
C :: CALL SOUND(B*38,C,3)::
NEXT I
2460 SUBEND
```

ACHTUNG!
089/1298011 Ab 1.10.85!

BOOGABOO

Fortsetzung von Seite 123

beiden LOCATE-Befehlen der Faktor vor der Statusvariablen ‚X' des Joysticks vergrößert werden. Eventuelle Verbesserungen oder Erweiterungen des Programmes können nach einer Freisetzung von Speicherplatz vorgenommen werden. Dazu kann vielleicht die Spielerläuterung ersatzlos gestrichen und das Titelbild dementsprechend abgeändert werden. Dem Bastler stehen hier alle Wege offen.

Aufbau des Programms BOOGABOO

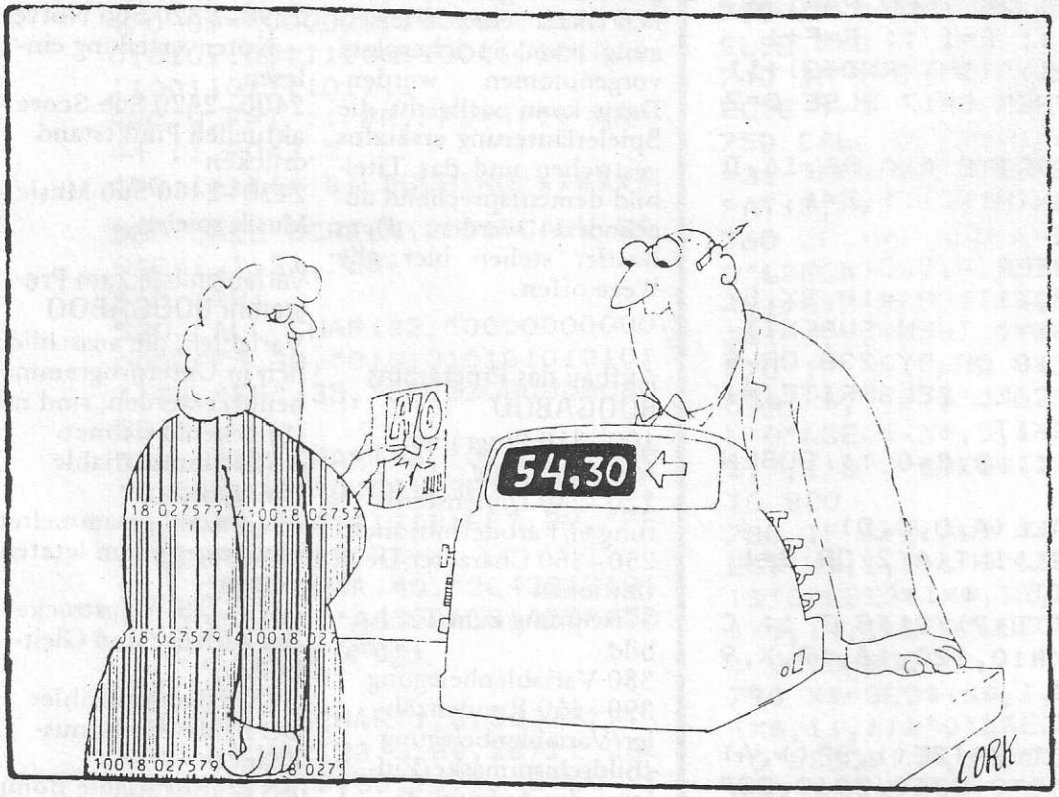
100–210 Programm Identifizierung
220–240 Dimensionierungen/Farbdefinitionen
250–360 Character-Definitionen
370 Sprung zum Titelbild
380 Variablenbelegung
390–450 Rundenzähler/Variablenbelegung /Bildschirmmaske/Einlesen der Aufzugs- & Noten-Daten
460–530 Hauptprogramm
540–610 Joystick-Abfrage/Bewegung
620–780 Titelbild
790–901 Absturz/ Zeit überschritten
920–980 Bonus
990–1100 Nächste Runde
1110–1170 Noten setzen
1180–1250 Noten sammeln
1260–1360 Spielende
1370–1530 Spielerläuterungen
1540–1600 Daten Aufzüge
1610–1800 Daten Display -1-2-3-
1810–1860 Daten Musik-Noten
1870–1930 Daten Musik
1940–1990 Sub Lift – Einlesen der Aufzugdaten
2000–2070 Sub Bild – Bildschirmmaske
2080–2190 Sub Mov – Aufzüge und Gleitböden bewegen
2200–2250 Sub Fns – Notenschlüssel setzen und bewegen
2260–2280 Sub Treff – Hindernisse löschen
2290–2320 Sub Notver – Notenverteilung einlesen
2400–2420 Sub Score – aktuellen Punktstand drucken
2430–2460 Sub Music – Musik spielen

Variablenliste zum Programm BOOGABOO

Variablen, die ausschließlich in Unterprogrammen benutzt werden, sind mit (S) gekennzeichnet.
A (S) Statusvariable Hindernis
AN Anzahl gesammelter Noten nach dem letzten Setzen
AW (..) (S) Wegstrecke der Aufzüge und Gleitböden
B (S) Hinderniszähler
BC Zähler für Bonuschancen
BN Statusvariable Bonus
Hi Highscore
I,J Schleifenvariablen
MA Restleben
NS Gesamtanzahl gesammelter Noten
PA (S) Gegenwärtiger Mustercode des Notenschlüssels
PU Punkte
P („) (S) Maximale Positionen der Aufzüge und Gleitböden
R Rundenzähler für Bildschirmmaske
R1 Absoluter Rundenzähler
SX/SY Punktzeile / -spalte der Spielerfigur
SX/SY (S) Punktzeile/ -spalte der Aufzüge der Gleitböden
TI Zeit
SP (..) Spalte für Notendruck
ZE(..) Zeile für Notendruck
X,Y Hebelpositionen des Joysticks

Peter Krawinkel

SUPER! DAS SUPER GRAFIK-PROGRAMM



Nach dem Starten des Programmes erscheint das Hauptmenue. Dort können sie zwischen 4 Möglichkeiten wählen. Wählt man Laden oder Speichern, dann wird man nach dem Namen und der Nummer der Datei gefragt. Wählt man Aufruf der Grafik, dann erscheint ein zweites Menue. Dort kann zwischen Erstellen und Definition gewählt werden. Wählt man Definition, so kann man dort mit Hilfe der Tastatur ein Zeichen definieren. Das Umrechnen in den HEX-Code macht der Rechner. Wählt man jedoch Erstellen, so kann man dort eine neue Grafik erstellen oder eine alte Grafik verbessern.

Beschreibung der einzelnen Programmteile:

Hauptprogramm: Hier werden die benötigten Variablen initialisiert.
LOES(chen): Hier wird der Grafikbildschirm gelöscht und gefragt, ob wirklich gelöscht werden soll (LO(esch)FR(age)).
SPRU(ng): Abfrage auf gedrückte Taste und Feststellung, ob eine richtige gedrückt wurde.
LADE: Lädt Grafik von Kassette oder Diskette (FRAG(e)).
SAVE: Speichert Grafik auf Kassette oder Diskette (FRAG(e)).
HAUP(tmenue): Erstellt Menue für das Hauptprogramm.
TITEL: Erstellt Titelbild.
LEER: Leert Bildschirm und initialisiert Farben neu.
GRAF(ik): Fragt, ob man eine Grafik erstellen oder ein Zeichen neu definieren will.
ENDE: Fragt, ob man wirklich beenden will.
ERST(elle): Wenn schon etwas definiert bzw. eine Grafik erstellt wurde, die alte Grafik erstellt (BILD).
Beim Drücken einer beliebigen Taste erscheint das Zeichen auf dem Bildschirm unterhalb des Cursors (DRUC(ke)), solange ihr ASCII-Code größer 39 und kleiner 144 ist.

Um den Cursor zu bewegen, müssen die Cursortasten in Verbindung mit FCTN gedrückt werden. Um die Farbe des Bildschirmes zu verändern, muß FCTN REDO drücken. Die Vordergrundfarbe einer Gruppe kann durch FCTN DEL und die Hintergrundfarbe durch FCTN INS erhöht werden. Um diese Veränderungen hervorzurufen, muß ein Zeichen dieser Gruppe unter dem Cursor sein. Mit FCTN BACK kann man dieses Unterprogramm verlassen. Mit FCTN AID erhält man das Menue zu diesem Unterprogramm.
SET (cursor): Zeigt, wo sich der Cursor befindet).
BEST(imme): Bestimmt, von welcher Zeichengruppe die Farbe verändert werden soll.
ERH(öhe): Erhöht oder vermindert einen Wert um 1 bis zu einer maximalen oder minimalen Grenze. Ist eine dieser Grenzen erreicht, so wird der Wert auf die andere Grenze gesetzt.
DEFI(nieren): Hier kann ein Zeichen mit einer 8x8 Matrix definiert werden. Nach der Definition wird die Frage gestellt, ob das Zeichen noch verändert werden soll, oder ob man en neues Zeichen definieren will.
D(efi)MEN(ue): Erstellt das Menue zum Unterprogramm DEFI.
UMRE(chnen): Rechnet die Matrix in den dazugehörigen Hex-Code um.

Bedeutung der wichtigsten Variablen:

ZE$( ): Enthält den Teil des Grafikbildschirms, der mit DISPLAY AT dargestellt werden kann.
CH$ ( ): Enthält die Hex-Codes der neu definierten Zeichen.
SP$ ( ): Enthält die Spalten, die nur mit HCHAR dargetellt werden können.
VO ( ): Enthält die Vordergrundfarben der Zeicengruppen.
HI ( ): Enthält die Hintergrundfarben der Zeichengruppen.
SC: Enthält die Bildschirmfarbe.
DE: Gibt an, ob eine alte Grafik im Speicher ist.
I,J: Zählvariablen
ME$: Gibt die erlaubten Tasten an.

```
10 ! **********************
11 ! *                    *
12 ! *    SUPER-GRAF I    *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *     Uwe Klosa      *
17 ! *                    *
18 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *  Disk-Drive oder   *
23 ! *   Kass.-Rekorder   *
24 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     7591 Bytes     *
28 ! *                    *
29 ! **********************
100 ON WARNING NEXT :: OPTIO
N BASE 1 :: ON BREAK NEXT ::
 CALL TITEL :: RANDOMIZE
110 DIM ZE$(24),CH$(104),SP$
(4),VO(14),HI(14):: DE=-1
120 CALL LOES(ZE$(),CH$(),SP
$(),VO(),HI(),DE,SC)
130 CALL HAUP
140 ME$="SLDEG" :: CALL SPRU
(ME$,K)
150 IF K=83 THEN CALL SAVE(Z
E$(),CH$(),SP$(),VO(),HI(),S
C)
160 IF K=76 THEN CALL LADE(Z
E$(),CH$(),SP$(),VO(),HI(),S
C,DE)
170 IF K=71 THEN CALL GRAF(Z
E$(),CH$(),SP$(),VO(),HI(),S
C,DE):: CALL DELSPRITE(#1)
180 IF K=68 THEN 120
190 IF K=69 THEN CALL ENDE
200 IF K=0 THEN 140 ELSE 130
210 SUB LOES(ZE$(),CH$(),SP$
(),VO(),HI(),DE,SC)
220 IF DE>-1 THEN CALL LOFR(
WE):: IF WE=0 THEN SUBEXIT
230 FOR I=1 TO 24 :: ZE$(I)=
RPT$(" ",28):: NEXT I :: FOR
 I=1 TO 4 :: SP$(I)=RPT$(" "
,24):: NEXT I
240 FOR I=1 TO 14 :: VO(I)=2
 :: HI(I)=1 :: NEXT I :: SC=
8 :: FOR I=1 TO 104 :: CH$(I
)="000"&RPT$(" ",16):: NEXT
I :: DE=0
250 SUBEND
260 SUB SPRU(ME$,K)
270 CALL KEY(3,K,S):: IF S=0
 THEN 270
280 FOR I=1 TO LEN(ME$):: IF
 K=ASC(SEG$(ME$,I,1))THEN SU
BEXIT
290 NEXT I :: CALL SOUND(10,
110,15):: K=0
300 SUBEND
310 SUB HAUP
320 CALL LEER :: RESTORE 340
330 FOR I=1 TO 5 :: READ A$
:: DISPLAY AT(2*I+6,5):A$ ::
 NEXT I
340 DATA S> GRAFIK SPEICHERN
,L> GRAFIK LADEN,G> AUFRUF D
ER GRAFIK,D> GRAFIK LOESCHEN
,E> ENDE
350 SUBEND
360 SUB SAVE(ZE$(),CH$(),SP$
(),VO(),HI(),SC)
370 CALL FRAG(NA$,NU):: OPEN
 #NU:NA$,OUTPUT,SEQUENTIAL,I
NTERNAL,FIXED 80
380 FOR I=1 TO 24 STEP 3 ::
PRINT #NU:ZE$(I),ZE$(I+1),ZE
$(I+2):: NEXT I
390 FOR I=1 TO 14 STEP 2 ::
PRINT #NU:VO(I),HI(I),VO(I+1
),HI(I+1):: NEXT I
400 PRINT #NU:SP$(1),SP$(2),
SP$(3),SP$(4),SC
410 FOR I=1 TO 104 STEP 8 ::
 PRINT #NU:CH$(I),CH$(I+1),C
H$(I+2),CH$(I+3),CH$(I+4),CH
$(I+5),CH$(I+6),CH$(I+7):: N
EXT I :: CLOSE #NU
420 SUBEND
430 SUB FRAG(NA$,NU)
440 CALL LEER :: DISPLAY AT(
1,1):"D(ISKETTE ODER K(ASSET
TE ?:" :: ACCEPT AT(1,28)VAL
IDATE("DKdk"):A$
450 IF A$="D" OR A$="d" THEN
 NA$="DSK1" ELSE NA$="CS1"
460 DISPLAY AT(3,1):"DATEINU
MMER ?:" :: ACCEPT AT(3,14)V
ALIDATE(DIGIT):NU :: IF NU<0
 OR NU>255 THEN 460
470 DISPLAY AT(5,1):"DATEINA
ME ?:" :: ACCEPT AT(5,13)VAL
IDATE(UALPHA,DIGIT)SIZE(10):
N$ :: NA$=NA$&"."&N$
480 SUBEND
490 SUB LADE(ZE$(),CH$(),SP$
(),VO(),HI(),SC,DE)
500 CALL FRAG(NA$,NU):: OPEN
 #NU:NA$,INPUT ,SEQUENTIAL,I
NTERNAL,FIXED 80
510 FOR I=1 TO 24 STEP 3 ::
INPUT #NU:ZE$(I),ZE$(I+1),ZE
$(I+2):: NEXT I
520 FOR I=1 TO 14 STEP 2 ::
INPUT #NU:VO(I),HI(I),VO(I+1
),HI(I+1):: NEXT I
530 INPUT #NU:SP$(1),SP$(2),
SP$(3),SP$(4),SC
```

```
540 FOR I=1 TO 104 STEP 8 ::
 INPUT #NU:CH$(I),CH$(I+1),C
H$(I+2),CH$(I+3),CH$(I+4),CH
$(I+5),CH$(I+6),CH$(I+7):: N
EXT I :: CLOSE #NU
550 SUBEND
560 SUB LEER
570 CALL CLEAR :: CALL SCREE
N(5):: FOR I=1 TO 8 :: CALL
COLOR(I,2,16):: NEXT I
580 SUBEND
590 SUB ENDE
600 CALL LEER :: DISPLAY AT(
1,1):"WOLLEN SIE WIRKLICH BE
ENDEN?"
610 CALL SPRU("JN",K):: IF K
=74 THEN END ELSE IF K=78 TH
EN SUBEXIT ELSE 610
620 SUBEND
630 SUB LOFR(WE)
640 CALL LEER :: DISPLAY AT(
1,1):"GRAFIK WIRKLICH LOESCH
EN ?"
650 CALL SPRU("JN",K):: IF K
=74 THEN WE=1 ELSE IF K=78 T
HEN WE=0 ELSE 650
660 SUBEND
670 SUB GMEN
680 CALL LEER :: RESTORE 700
690 FOR I=1 TO 3 :: READ A$
:: DISPLAY AT(2*I+8,5):A$ ::
 NEXT I
700 DATA D> ZEICHENDEFINITIO
N,G> GRAFIK ERSTELLEN,Q> QUI
T
710 SUBEND
720 SUB GRAF(ZE$(),CH$(),SP$
(),VO(),HI(),SC,DE)
730 CALL CHARSET :: CALL GME
N :: CALL SPRU("DGQ",K):: IF
 K=68 THEN CALL DEFI(CH$()):
: DE=1 :: CALL DELSPRITE(#1)
740 IF K=71 THEN CALL ERST(Z
E$(),CH$(),SP$(),VO(),HI(),S
C,DE):: CALL DELSPRITE(#1)::
 DE=1 ELSE IF K=81 THEN SUBE
XIT
750 GOTO 730
760 SUBEND
770 SUB ERST(ZE$(),CH$(),SP$
(),VO(),HI(),SC,DE)
780 X=16 :: Y=12 :: CALL CHA
R(33,"00003C3C3C3C"):: CALL
BILD(ZE$(),CH$(),SP$(),VO(),
HI(),SC,DE)
790 CALL SETC(X,Y,SC):: CALL
 KEY(0,K,S):: IF S=0 THEN 79
0
800 IF K=3 THEN CALL BEST(I,
X,Y):: CALL ERH(VO(I),1,16):
: CALL COLOR(I,VO(I),HI(I))E
LSE IF K=4 THEN CALL BEST(I,
X,Y):: CALL ERH(HI(I),1,16):
: CALL COLOR(I,VO(I),HI(I))
810 IF K=6 THEN CALL ERH(SC,
1,16):: CALL SCREEN(SC)ELSE
IF K=15 THEN SUBEXIT ELSE IF
 K=1 THEN CALL EMEN(ZE$(),CH
$(),VO(),HI(),SC)
820 IF K=8 THEN X=-X :: CALL
 ERH(X,1,32)ELSE IF K=9 THEN
 CALL ERH(X,1,32)
830 IF K=10 THEN CALL ERH(Y,
1,24)ELSE IF K=11 THEN Y=-Y
:: CALL ERH(Y,1,24)
840 IF K>39 AND K<144 THEN C
ALL DRUC(K,X,Y,ZE$(),SP$()):
: GOTO 790 ELSE 790
850 SUBEND
860 SUB ERH(I,MI,MA)
870 IF I>0 THEN D=1 ELSE D=-
1 :: I=ABS(I)
880 I=I+D :: IF I<MI THEN I=
MA ELSE IF I>MA THEN I=MI
890 SUBEND
900 SUB SETC(X,Y,SC)
910 CALL COLOR(#1,1):: X1=X*
8-7 :: Y1=Y*8-7 :: IF SC>2 T
HEN FA=2 ELSE FA=16
920 CALL SPRITE(#1,33,FA,Y1,
X1)
930 SUBEND
940 SUB DRUC(K,X,Y,ZE$(),SP$
())
950 CALL HCHAR(Y,X,K):: IF X
<3 THEN SP$(X)=SEG$(SP$(X),1
,Y-1)&CHR$(K)&SEG$(SP$(X),Y+
1,24-Y)
960 IF X>30 THEN SP$(X-28)=S
EG$(SP$(X-28),1,Y-1)&CHR$(K)
&SEG$(SP$(X-28),Y+1,24-Y)
970 IF X>2 AND X<31 THEN ZE$
(Y)=SEG$(ZE$(Y),1,X-3)&CHR$(
K)&SEG$(ZE$(Y),X-1,30-X)
980 SUBEND
990 SUB BEST(A,X,Y)
1000 CALL GCHAR(Y,X,B):: A=I
NT(B/8)-3
1010 SUBEND
1020 SUB DEFI(CH$())
1030 ZE=33 :: CALL CHAR(33,"
FF8181818181FF",34,"FFFFFF
FFFFFFFFFF"):: X=13 :: Y=1
1040 CALL LEER :: FOR I=1 TO
 8 :: FOR J=1 TO 8 :: CALL H
CHAR(I,J+12,ZE):: NEXT J ::
NEXT I :: CALL DMEN
1050 CALL SETC(X,Y,2):: CALL
 SPRU("1OESDXUQ",K):: IF K=4
9 THEN ZE=34 ELSE IF K=48 TH
EN ZE=33 ELSE IF K=81 THEN S
UBEXIT
```

```
1060 IF K=69 THEN Y=-Y :: CA
LL ERH(Y,1,8)ELSE IF K=88 TH
EN CALL ERH(Y,1,8)ELSE IF K=
83 THEN X=-X :: CALL ERH(X,1
3,20)ELSE IF K=68 THEN CALL
ERH(X,13,20)
1070 IF K=85 THEN CALL UMRE(
CH$())ELSE CALL HCHAR(Y,X,ZE
):: GOTO 1050
1080 DISPLAY AT(20,1):"VERAE
NDERUNGEN VORNEHMEN ?" :: CA
LL SPRU("JN",K):: IF K=74 TH
EN CALL DMEN :: GOTO 1050 EL
SE IF K=0 THEN 1080
1090 DISPLAY AT(22,1):"NEUES
 ZEICHEN ?" :: CALL SPRU("JN
",K):: IF K=74 THEN 1030 ELS
E IF K=0 THEN 1090
1100 SUBEND
1110 SUB UMRE(CH$())
1120 DISPLAY AT(18,1):"WELCH
ES ZEICHEN(ASCII) ?" :: ACCE
PT AT(18,26)VALIDATE(DIGIT)S
IZE(3):ZE :: IF ZE<40 OR ZE>
143 THEN 1120
1130 CH$(ZE-39)=RPT$("0",3-L
EN(STR$(ZE)))&STR$(ZE)&""
1140 FOR I=1 TO 8 :: C$="" :
: FOR J=13 TO 16 :: CALL GCH
AR(I,J,G):: C$=C$&CHR$(G)::
NEXT J
1150 CALL WERT(C$,CH$(ZE-39)
):: C$="" :: FOR J=17 TO 20
:: CALL GCHAR(I,J,G):: C$=C$
&CHR$(G):: NEXT J
1160 CALL WERT(C$,CH$(ZE-39)
):: NEXT I :: DISPLAY AT(12,
1):SEG$(CH$(ZE-39),4,16)
1170 SUBEND
1180 SUB WERT(C$,D$)
1190 E=0 :: FOR I=0 TO 3 ::
IF SEG$(C$,4-I,1)=CHR$(34)TH
EN D=2^I ELSE D=0
1200 E=E+D :: NEXT I :: H$="
0123456789ABCDEF"
1210 D$=D$&SEG$(H$,E+1,1)
1220 SUBEND
1230 SUB BILD(ZE$(),CH$(),SP
$(),VO(),HI(),SC,DE)
1240 CALL CLEAR :: IF DE=0 T
HEN 1280
1250 FOR I=1 TO 104 :: V=VAL
(SEG$(CH$(I),1,3)):: IF V>39
 THEN CALL CHAR(V,SEG$(CH$(I
),4,16))
1260 NEXT I :: FOR I=1 TO 24
 :: DISPLAY AT(I,1):ZE$(I)::
 NEXT I
1270 FOR I=1 TO 2 :: FOR J=1
 TO 24 :: CALL HCHAR(J,I,ASC
(SEG$(SP$(I),J,1))):: CALL H
CHAR(J,I+30,ASC(SEG$(SP$(I+2
),J,1))):: NEXT J :: NEXT I
1280 CALL SCREEN(SC):: FOR I
=1 TO 14 :: CALL COLOR(I,VO(
I),HI(I)):: NEXT I
1290 SUBEND
1300 SUB DMEN
1310 RESTORE 1330
1320 FOR I=1 TO 4 :: READ A$
 :: DISPLAY AT(I*2+16,1):A$
:: NEXT I
1330 DATA O> PUNKT AUS...1>
PUNKT EIN,E> HOCH.....X>
RUNTER,S> LINKS.....D> REC
HTS
1340 DATA U> UMRECHNEN...Q>
QUIT
1350 SUBEND
1360 SUB TITEL
1370 CALL CLEAR :: CALL SCRE
EN(2):: FOR I=1 TO 14 :: CAL
L COLOR(I,INT(RND*13+3),2)::
 NEXT I
1380 FOR I=1 TO 32 :: CALL H
CHAR(1,I,42):: CALL HCHAR(24
,I,42):: NEXT I
1390 FOR I=1 TO 24 :: CALL H
CHAR(I,1,42):: CALL HCHAR(I,
32,42):: NEXT I :: DISPLAY A
T(5,9):"TERRA - SOFT" :: DIS
PLAY AT(7,9):"Praesentiert"
1400 DISPLAY AT(10,8):"SUPER
 - GRAF I" :: DISPLAY AT(11,
8):"==============" :: DISPL
AY AT(14,1):"fuer TI99/4A +
EXT + Kas/Dis"
1410 DISPLAY AT(16,1):"Idee
& Ausfuehrung:Uwe Klosa" ::
DISPLAY AT(20,5):"(c) Uwe Kl
osa   1985"
1420 FOR I=1 TO 10 :: FOR J=
1 TO 14 :: CALL COLOR(J,INT(
RND*13+3),2):: NEXT J :: NEX
T I
1430 SUBEND
1440 SUB EMEN(ZE$(),CH$(),VO
(),HI(),SC)
1450 RESTORE 1460 :: CALL CH
ARSET :: CALL SCREEN(16):: F
OR I=1 TO 5 :: READ A$ :: DI
SPLAY AT(I+19,1):A$ :: NEXT
I
1460 DATA FCTN E>HOCH // FCT
N X>RUNTER,FCTN S>LINKS / FC
TN D>RECHTS,FCTN 1>VOR+1 //
FCTN 2>HIN+1
1470 DATA FCTN 8>SCFAR+1 / F
CTN 9>QUIT,FCTN 7>AUFRUF DIE
SES MENUES
1480 CALL KEY(0,K,S):: IF S=
0 THEN 1480
```

```
1490 FOR I=1 TO 56 :: V=VAL(
SEG$(CH$(I),1,3)):: IF V>39
THEN CALL CHAR(V,SEG$(CH$(I)
,4,16))
1500 NEXT I :: FOR I=1 TO 14
 :: CALL COLOR(I,VO(I),HI(I)
):: NEXT I :: CALL SCREEN(SC
):: CALL CHAR(33,"00003C3C3C
3C")
1510 FOR I=19 TO 24 :: DISPL
AY AT(I,1):ZE$(I):: NEXT I
1520 SUBEND
```

DER TI ALS SCHREIB-MASCHINEN LEHRER

Das Programm Touch Typing Teacher ist ein Schreibmaschinenlehrgang für den TI99/4A.

Es nutzt fast allen graphischen Möglichkeiten des TI99/4A und auch der Ton kommt nicht zu kurz. Wenn man intensiv mit dem Programm arbeitet, kann man nach einiger Zeit die Tastatur des TI fast blind beherrschen, aber auch wenn man sich nicht so intensiv mit dem Programm beschäftigt, ist es nicht abzustreiten, daß man die Tastatur wesentlich besser beherrscht als vorher und sich auch die Schreibgeschwindigkeit stark gesteigert hat. Das Programm stellt also eine wirkliche Alternative zu dem von Texas Instruments vertriebenen Programm Touch Typing Tutor dar.

Der Benutzer kann selbst entscheiden, wann er genug von einer Übung hat, oder, wenn er glaubt, eine Übung gut genug zu beherrschen, kann er sich einfach eine andere Übung heraussuchen.

Aber natürlich kann man mit dem Programm nicht nur üben, man kann z.B. auch auf Zeit schreiben (der Computer mißt die Zeit), oder Texte nachschreiben, die der Computer dann auf ihre Richtigkeit (orthographische) überprüft.

Das Programm besitzt eine benutzerfreundliche Menütechnik, die es jedem Benutzer kinderleicht macht, mit dem Programm umzugehen.

Nach dem Laden des Programms und Eingabe von RUN oder vor dem Laden von RUN „CS1" erscheint nach ein paar Sekunden das erste Titelbild. Nach Drücken einer Taste wird der Bildschirm schwarz, und nach cirka 15 Sekunden erscheint das eigentliche Titelbild, mit der Tastatur des Ti99 /4A, auf dem Bildschirm. Wenn Sie nun eine Taste drücken, erscheint das Hauptmenü. Das Hauptmenü umfaßt 6 Wahlmöglichkeiten. Wenn Sie aus irgendeinem Unterprogramm wieder zum Hauptmenü zurückkehren wollen, dann müssen Sie nur bei einer CALL KEY Abfrage FCTN 9 eingeben (drücken).

Wenn Sie nun Punkt 1 drücken, können Sie den Umgang mit der Tastatur etwas üben. Dazu druckt der Computer die Tastatur auf dem Bildschirm aus. Dann wird kurz erklärt, mit welchen Tasten Sie in welches Unterprogramm kommen. Mit FCTN 8 können Sie den Computer dazu veranlassen, daß er die einzelnen Symbole auf die Tastatur druckt, da diese ja nicht vorhanden sind. Sind die Symbole gedruckt, können Sie die Symbole wieder löschen. Dieses Unterprogramm dient nur als Gedankenstütze. Mit FCTN 9 können Sie zum Hauptmenü zurückkehren. Wenn Sie aber nun irgendeine andere Taste drücken, beginnt der Computer mit dem „Kennenlernen der Tastatur". Auf dem Bildschirm ist nun die Tastatur des Ti99/4A zu sehen, allerdings ohne Symbole. Wenn Sie nun irgendeine Taste drücken, erscheint das Symbol, welches auf der Taste zu sehen ist, auf der entsprechenden Taste auf dem Bildschirm und verschwindet kurz danach wieder. Wollen Sie dieses Symbol also länger sehen, müssen Sie nur den Finger länger auf der entsprechenden Taste lassen.

Mit diesem Unterprogramm können Sie lernen, wo die einzelnen Tasten liegen. Wie schon beschrieben, können Sie mit FCTN 9 zurück ins Hauptmenü und mit FCTN 8 die Symbole auf die Tasten drucken lassen. Bei diesem Unterprogramm und bei allen anderen sollten Sie versuchen, nicht auf die Tastatur zu sehen, da sonst der Lernerfolg nicht sehr groß ist.

Folgende Symbole können durch Drücken der entsprechenden Taste nicht auf den Bildschirm gebracht werden (gilt nur für 1 und 2 aus dem Hauptmenü):
alle Kleinbuchstaben (gilt auch für 3 und 4)
die Zeichen Links geschweifte Klammer, Doppelstrich, Rechts geschweifte Klammer, Tilde, Del

Wenn Sie eines dieser Zeichen eingeben, ertönt ein Warnton (nur bei 1 und 2), bei 3 und 4 erscheinen andere Zeichen, da die ASCII-Codes dieser Zeichen mit anderen Zeichen belegt wurden (CALL CHAR).

Wenn Sie 2 drücken, kommen Sie in ein Unterprogramm, welches überprüft, wie gut Sie sich auf der TI-Tastatur auskennen. Auf dem Bildschirm erscheint nun wiederum die TI-Tastatur ohne Symbole. Allerdings erscheint nun immer ein Symbol auf der Bildschirmtastatur, welches Sie auf der TI-Tastatur finden müssen (blind, ohne auf die Tastatur zu sehen). Wenn Sie die richtige Taste gedrückt haben, ertönt eine kurze „Siegesmelodie" und es erscheint ein neues Symbol. Wenn Sie allerdings falsch gedrückt haben, erscheint das falsche Symbol nur für kurze Zeit auf der Bildschirmtastatur und es erklingt eine kleine Melodie, die den falschen Tastendruck akustisch erkennen läßt. Falls Sie eine der nicht erlaubten Tasten gedrückt haben (siehe bei 1 alle Zeichen mit einem ASC II Code über 96), dann ertönt ein kurzer Warnton.

Wenn sie nun einige Zeit mit dem Unterprogramm gearbeitet haben und Sie wollen zum Hauptmenü zurückkehren, müssen Sie ebenfalls

Bitte lesen Sie weiter auf Seite 133

```
10 ! ************************
11 ! *                      *
12 ! *TOUCH TYPING TEACHER*
13 ! *                      *
14 ! *    Copyright by      *
15 ! *   Dirk Junghans      *
16 ! *                      *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole   *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    10186 Bytes       *
28 ! *                      *
29 ! ************************
30 !
100 OPTION BASE 1 :: GOTO 22
0
110 CALL CHAR :: CALL HCHAR
:: CALL CLEAR :: CALL SCREEN
 :: CALL COLOR :: CALL CHARS
ET :: CALL SOUND :: CALL KEY
120 CALL TASTA :: CALL SONDE
R :: CALL ANZ :: CALL EING :
: CALL ABFEING
130 JN$=""
140 I,S,SE,WA,DJ,DD,I,K,S,X1
,Y1,ZAE,I1,I2,TI
150 DATA ABC
160 !@P-
170 !*bei fragen:          *
180 !*dirk junghans        *
190 !*am fort-biehler 9    *
200 !*6503 mz-kastel       *
210 !************************
220 CALL CLEAR
230 CALL SCREEN(10)
240 FOR I=2 TO 8 :: CALL COL
OR(I,5,10):: NEXT I
250 CALL COLOR(13,2,10,14,2,
10)
260 CALL CHAR(131,"00"&RPT$(
"01",7),132,"FC8485858484868
2",133,"0080404080000000C")
270 CALL CHAR(134,"01FF80C04
060381C",135,"8C9921213D0505
05",136,"12BA8A8ABAA1A1A1")
280 CALL CHAR(137,"0E0301",1
38,"C4E23110180C0703",139,"2
24C90204040A0E0")
290 CALL CLEAR
300 DISPLAY AT(1,1):"DJIS" :
: DISPLAY AT(3,6):"EIN PROGR
AMM VON" :: DISPLAY AT(5,8):
"DIRK JUNGHANS"
310 DISPLAY AT(8,5):"FUER DE
N TI99/4A VON" :: DISPLAY AT
(16,6):"TEXAS INSTRUMENTS"
320 DISPLAY AT(19,1):"DAS PR
OGRAMM WURDE DER.....KLASSE
 10B DER HLS WIESBADENGEWIDM
ET."
330 DISPLAY AT(11,13):CHR$(1
31);CHR$(132);CHR$(133):: DI
SPLAY AT(12,13):CHR$(134);CH
R$(135);CHR$(136)
340 DISPLAY AT(13,13):CHR$(1
37);CHR$(138);CHR$(139)
350 DISPLAY AT(24,3):"PRESS
A KEY TO CONTINUE"
360 CALL KEY(0,K,S):: IF S=0
 THEN 360
370 CALL CLEAR :: CALL CHARS
ET :: ON BREAK NEXT
380 CALL SCREEN(2)
390 FOR I=0 TO 14 :: CALL CO
LOR(I,2,2):: NEXT I
400 !@P+
410 DIM M(5),X(64),Y(64),EI$
(10),NA$(10)
420 !@P-
430 RESTORE 1920 :: FOR I=1
TO 64 :: READ Y1,X1 :: Y(I)=
Y1 :: X(I)=X1 :: NEXT I
440 CALL CHAR(104,RPT$("0",1
5)&"1",105,RPT$("0",12)&"FF0
0",106,RPT$("0",14)&"81")
450 CALL CHAR(107,RPT$("0",1
4)&"80",108,"0305"&RPT$("09"
,6),109,"C3A5"&RPT$("9",12))
460 CALL CHAR(110,"C0A090909
0909090",111,"090A0C0F",112,
"FF0000FF",113,"995A3CFF",11
4,"905030F0")
470 CALL CHAR(101,"0000EA85E
585E5",97,"304878CA2020203A"
,102,"00E384E8888483",99,"00
374282824232",115,"748484661
515E5")
480 M(1)=3 :: M(2)=4 :: M(3)
=5 :: M(4)=4 :: M(5)=4
490 CALL TASTA(M()):: CALL S
ONDER :: CALL ANZ(X(),Y()):
 DISPLAY AT(2,4):"TOUCH TYPI
NG TEACHER"
500 DISPLAY AT(21,2):"(C) 19
85 BY DIRK JUNGHANS"
510 DISPLAY AT(24,4):"PRESS
ANY KEY TO BEGIN"
520 CALL SCREEN(16)
530 FOR I=0 TO 8 :: CALL COL
OR(I,2,16):: NEXT I
540 FOR I=9 TO 12 :: CALL CO
LOR(I,5,16):: NEXT I
550 CALL KEY(0,K,S):: IF S=0
 THEN 550
560 CALL CLEAR
570 DISPLAY AT(1,1):"DJIS"
580 DISPLAY AT(5,1):"1 =>KEN
NENLERNEN MIT DER.....TAS
TATUR"
590 DISPLAY AT(8,1):"2 =>WIE
```

```
 1 JEDOCH MIT.............VOR
GEGEBENEN ZEICHEN"
600 DISPLAY AT(11,1):"3 =>NA
CHSCHREIBEN VORHER.........EI
NGEGEBENENER TEXTE"
610 DISPLAY AT(14,1):"4 =>SC
HREIBEN AUF ZEIT VON.......FR
EIEN TEXTEN"
620 DISPLAY AT(17,1):"5 =>DR
UCKEN DER TASTATUR AUF....BI
LDSCHIRM"
630 DISPLAY AT(20,1):"6 =>EN
DE"
640 DISPLAY AT(24,4):"CHOOSE
 A KEY TO BEGIN"
650 CALL KEY(0,K,S):: IF S=0
 THEN 650
660 IF K<49 OR K>54 THEN 650
670 ON K-48 GOTO 740,1470,85
0,1570,1850,680
680 CALL CLEAR
690 DISPLAY AT(1,1):"DJIS" :
: DISPLAY AT(10,2):"WOLLEN S
IE EIN NEUES"
700 DISPLAY AT(11,2):"PROGRA
MM LADEN ???"
710 DISPLAY AT(15,10):"(J/N)
" :: ACCEPT AT(15,16)BEEP VA
LIDATE("J","N")SIZE(1):JN$
720 IF JN$="N" THEN STOP
730 CALL CHARSET :: CALL CLE
AR :: RUN "CS1"
740 CALL CLEAR
750 CALL TASTA(M())
760 CALL SONDER
770 DISPLAY AT(1,1):"DJIS" :
: DISPLAY AT(3,2):"NR. 1"
780 DISPLAY AT(20,2):"PRESS
FCTN 8 TO PRINT KEYS"
790 DISPLAY AT(21,2):"PRESS
FCTN 9 TO RETURN"
800 DISPLAY AT(24,2):"PRESS
NOW A KEY TO GO ON"
810 CALL KEY(0,K,S):: IF S=0
 THEN 810
820 DISPLAY AT(20,1):RPT$(""
,28):: DISPLAY AT(21,1):RPT$
("",28):: DISPLAY AT(24,1):R
PT$("",28)
830 CALL EING(Y(),X())
840 GOTO 560
850 CALL CLEAR :: ZAE=0
860 PRINT "JETZT KOENNEN SIE
 EINEN":"VON IHNEN EINGEGEBE
N TEXT":"NACHSCHREIBEN.ZUERS
T,MUESSEN"
870 PRINT "SIE EINEN 10 ZEIL
IGEN TEXT":"EINGEBEN.DANACH
ERSCHEINT"
880 PRINT "DER TEXT IMMER ZE
ILE FUER":"ZEILE AUF DEM BIL
DSCHIRM,"
890 PRINT "DIE SIE DANN NACH
SCHREIBEN":"MUESSEN.": : : :
900 PRINT TAB(2):"PRESS ANY
KEY TO CONTINUE"
910 DISPLAY AT(1,1):"DJIS":
: DISPLAY AT(3,2):"NR.3"
920 CALL KEY(0,K,S):: IF S=0
 THEN 920 :: IF K=15 THEN 56
0
930 CALL CLEAR :: DISPLAY AT
(1,1):"DJIS"
940 DISPLAY AT(3,5):"EINGABE
 DES TEXTES"
950 DISPLAY AT(5,1):"START"
:: DISPLAY AT(17,25):"ENDE"
960 DISPLAY AT(12,3):"PRESS
ANY KEY TO START"
970 CALL KEY(0,K,S):: IF S=0
 THEN 970 :: IF K=15 THEN 56
0
980 DISPLAY AT(12,3):RPT$(""
,22)
990 FOR I=1 TO 10
1000 ACCEPT AT(I+5,1):EI$(I)
1010 NEXT I
1020 DISPLAY AT(20,1):"DIE E
INGABE IST BEENDET"
1030 DISPLAY AT(24,2):"PRESS
 ANY KEY TO CONTINUE"
1040 FOR I=1 TO 50 :: NEXT I
1050 CALL KEY(0,K,S):: IF S=
0 THEN 1050 :: IF K=15 THEN
560
1060 CALL CLEAR :: ZAE=0
1070 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 2"
1080 DISPLAY AT(10,2):"r DIE
SEN TEXT NACHSCHREIBEN"
1090 DISPLAY AT(6,1):RPT$("-
",28): :RPT$("-",28):: DISPL
AY AT(11,1):RPT$("=",28): :R
PT$("=",28)
1100 FOR I=1 TO 10
1110 DISPLAY AT(12,1)BEEP:EI
$(I)
1120 ACCEPT AT(7,1):NA$(I)
1130 NEXT I
1140 DISPLAY AT(15,6):"BITTE
 WARTEN !!!"
1150 FOR I=1 TO 10
1160 IF NA$(I)<>EI$(I)THEN G
OSUB 1430
1170 NEXT I
1180 DISPLAY AT(18,2):"SIE M
ACHTEN ";ZAE;" FEHLER"
1190 DISPLAY AT(19,2):"BEIM
ABSCHREIBEN"
1200 DISPLAY AT(22,1):"1 =>D
RUCKEN DER BEIDEN TEXTE"
1210 DISPLAY AT(23,1):"2 =>N
```

```
OCHMAL NACHSCHREIBEN"
1220 DISPLAY AT(24,1):"FCTN
9 =>ZURUECK ZUM MENUE"
1230 CALL KEY(0,K,S):: IF S=
0 THEN 1230 :: IF K=15 THEN
560
1240 IF K=49 THEN 1260
1250 IF K=50 THEN 1060 ELSE
1230
1260 CALL CLEAR
1270 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 2"
1280 DISPLAY AT(24,4):"PRESS
 A KEY TO PRINT"
1290 CALL KEY(0,K,S):: IF S=
0 THEN 1290 :: IF K=15 THEN
560
1300 DISPLAY AT(4,1):"EINGAB
ETEXT"
1310 FOR I=1 TO 10 :: DISPLA
Y AT(I+5,1):EI$(I):: NEXT I
1320 DISPLAY AT(19,1):RPT$("
-",28): :RPT$("-",28)
1330 DISPLAY AT(18,1):"NACHS
CHREIBE TEXT"
1340 DISPLAY AT(24,2):"PRESS
 ANY KEY TO CONTINUE"
1350 FOR I=1 TO 10
1360 DISPLAY AT(20,1):NA$(I)
1370 FOR I2=1 TO 50 :: NEXT
I2
1380 CALL KEY(0,K,S):: IF S=
0 THEN 1370 :: IF K=15 THEN
560
1390 NEXT I
1400 DISPLAY AT(23,1):"1 =>N
OCHMAL NACHSCHREIBEN"
1410 DISPLAY AT(24,1):"FCTN
9 =>ZURUECK ZUM MENUE"
1420 CALL KEY(0,K,S):: IF S=
0 THEN 1420 :: IF K=15 THEN
560 :: IF K=49 THEN 1060 ELS
E 1420
1430 FOR I1=1 TO 28
1440 IF SEG$(NA$(I),I1,1)<>S
EG$(EI$(I),I1,1)THEN ZAE=ZAE
+1
1450 NEXT I1
1460 RETURN
1470 CALL CLEAR
1480 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 2"
1490 CALL TASTA(M())
1500 CALL SONDER
1510 DISPLAY AT(20,2):"PRESS
 FCTN 9 TO RETURN"
1520 DISPLAY AT(24,2):"PRESS
 NOW A KEY TO GO ON"
1530 CALL KEY(0,K,S):: IF S=
0 THEN 1530
1540 DISPLAY AT(20,1):RPT$("
",28):: DISPLAY AT(24,1):RPT
$("",28)
1550 CALL ABFEING(Y(),X())
1560 GOTO 560
1570 CALL CLEAR
1580 TI=0
1590 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 4"
1600 DISPLAY AT(6,1):"BITTE
DRUECKEN SIE EINE":"TASTE,DA
NN BEGINNT DER":"ZAEHLER ZU
ZAEHLEN"
1610 DISPLAY AT(9,1):"UND SI
E KOENNEN BEGINNEN":"MIT DEM
 SCHREIBEN.":"VIEL GLUECK !"
1620 DISPLAY AT(13,1):"SIE M
UESSEN 168 ZEICHEN":"SCHREIB
EN.DIE SCHWARZEN"
1630 DISPLAY AT(15,1):"KAEST
CHEN WERDEN AM ENDE":"NICHT
MITGEZAEHLT."
1640 DISPLAY AT(17,1):"SIE K
OENNEN AUCH KEINE":"KORREKTU
REN AM TEXT VOR-"
1650 DISPLAY AT(19,1):"NEHME
N.SCHREIBEN SIE DES-":"HALB
BEI EINEM FEHLER":"WEITER."
1660 DISPLAY AT(24,3):"PRESS
 ANY KEY TO START"
1670 CALL KEY(0,K,S):: IF S=
0 THEN 1670 :: CALL CLEAR
1680 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 4"
1690 DISPLAY AT(4,1):"START"
 :: DISPLAY AT(11,25):"ENDE"
1700 S=0 :: SE=5 :: WA=3
1710 CALL KEY(0,K,S):: TI=TI
+1 :: IF S=0 THEN 1710
1720 CALL HCHAR(SE,WA,K):: C
ALL HCHAR(SE,WA+1,30)
1730 WA=WA+1
1740 IF WA>30 THEN WA=3 :: S
E=SE+1 ELSE 1750
1750 IF NOT SE>10 THEN GOTO
1710
1760 DISPLAY AT(17,2):"SIE B
RAUCHTEN FUER 168"
1770 DISPLAY AT(18,2):"ZEICH
EN ";TI
1780 DISPLAY AT(19,2):"ZEITE
INHEITEN."
1790 DISPLAY AT(20,2):"NICHT
 SCHLECHT !!!"
1800 DISPLAY AT(23,2):"1 =>N
OCH EINMAL"
1810 DISPLAY AT(24,2):"FCTN
9 =>ZURUECK ZUM MENUE"
1820 FOR I=1 TO 100 :: NEXT
I
1830 CALL KEY(0,K,S):: IF S=
0 THEN 1820 :: IF K=49 THEN
```

```
1570 :: IF K<>15 THEN 1830
1840 GOTO 560
1850 CALL CLEAR
1860 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 5"
1870 CALL TASTA(M())
1880 CALL SONDER
1890 CALL ANZ(X(),Y())
1900 DISPLAY AT(24,4):"PRESS
 A KEY TO RETURN"
1910 CALL KEY(0,K,S):: IF S=
0 THEN 1910 :: GOTO 560
1920 DATA 6,4,9,23,6,8,6,10,
6,12,6,16,9,21,6,20,6,22,6,1
8,6,24,15,21,9,25,15,23,9,25
,6,22
1930 DATA 6,4,6,6,6,8,6,10,6
,12,6,14,6,16,6,18,6,20,12,2
4,12,24,15,21,6,24,15,23
1940 DATA 9,19,6,6,12,6,15,1
5,15,11,12,10,9,9,12,12,12,1
4,12,16,9,19,12,18,12,20
1950 DATA 12,22,15,19,15,17,
9,21,9,23,9,5,9,11,12,8,9,13
,9,17,15,13,9,7,15,9,9,15,15
,7
1960 DATA 9,11,15,7,9,13,6,1
4,9,17,15,11
1970 !@P+
1980 SUB TASTA(M()):: DJ=5
1990 FOR I=1 TO 5
2000 DISPLAY AT(DJ,M(I)):"h"
&RPT$("ij",10)&"ik"
2010 DISPLAY AT(DJ+1,M(I)):"
l"&RPT$(" m",10)&" n"
2020 DISPLAY AT(DJ+2,M(I)):"
o"&RPT$("pq",10)&"pr"
2030 DJ=DJ+3
2040 NEXT I
2050 CALL HCHAR(17,11,105,15
):: CALL HCHAR(19,11,112,15)
:: CALL HCHAR(18,11,32,15)
2060 CALL HCHAR(14,28,105)::
 CALL HCHAR(14,29,107):: CAL
L HCHAR(15,28,32):: CALL HCH
AR(15,29,110)
2070 CALL HCHAR(16,28,112)::
 CALL HCHAR(16,29,114)
2080 SUBEND
2090 SUB ABFEING(Y(),X())
2100 DD=INT((96-32+1)*RND)+3
2
2110 IF DD=32 THEN GOSUB 226
0 :: GOTO 2130 ELSE 2120
2120 CALL HCHAR(Y(DD-32),X(D
D-32)+2,DD)
2130 CALL KEY(0,K,S):: IF S=
0 THEN 2130
2140 IF K=15 THEN SUBEXIT
2150 IF K>96 OR K<32 THEN CA
LL SOUND(100,300,3,-7,3):: G
OTO 2130 :: ELSE 2160
2160 IF K<>DD AND K=32 THEN
GOSUB 2270 :: GOTO 2240 ELSE
 IF K<>DD THEN 2210
2170 CALL SOUND(100,392,0)::
 CALL SOUND(100,392,0,330,0)
:: CALL SOUND(100,392,0,330,
0):: CALL SOUND(200,440,0,39
2,0)
2180 CALL SOUND(200,392,0,33
0,0)
2190 IF DD<>32 THEN 2200 ::
CALL HCHAR(18,15,32,5):: GOT
O 2100
2200 CALL HCHAR(Y(DD-32),X(D
D-32)+2,32):: GOTO 2100
2210 CALL HCHAR(Y(K-32),X(K-
32)+2,K)
2220 FOR I=1 TO 30 :: NEXT I
2230 CALL HCHAR(Y(K-32),X(K-
32)+2,32):: CALL HCHAR(Y(DD-
32),X(DD-32)+2,DD)
2240 CALL SOUND(100,220,0)::
 CALL SOUND(100,175,0):: CAL
L SOUND(200,-3,0)
2250 GOTO 2130
2260 CALL HCHAR(18,15,30,5):
: RETURN
2270 CALL HCHAR(18,15,30,5)
2280 FOR I=1 TO 30 :: NEXT I
2290 CALL HCHAR(18,15,32,5):
: RETURN
2300 SUBEND
2310 SUB ANZ(X(),Y())
2320 FOR I=12 TO 59 :: IF I+
32>90 OR I+32<44 OR I+32=45
OR I+32=58 OR I+32=60 OR I+3
2=62 OR I+32=63 OR I+32=64 T
HEN 2340
2330 CALL HCHAR(Y(I),X(I)+2,
I+32)
2340 NEXT I
2350 SUBEND
2360 SUB SONDER
2370 CALL HCHAR(15,27,115)::
 CALL HCHAR(12,28,101)
2380 CALL HCHAR(15,7,115)::
CALL HCHAR(18,7,97):: CALL H
CHAR(18,9,99):: CALL HCHAR(1
8,27,102)
2390 SUBEND
2400 SUB EING(Y(),X())
2410 CALL KEY(0,K,S):: IF S=
0 THEN 2410 :: IF K<>6 THEN
2440
2420 CALL ANZ(X(),Y())
2430 GOSUB 2510 :: GOTO 2410
2440 IF K=15 THEN SUBEXIT
2450 IF K<>32 THEN 2460 ELSE
 CALL HCHAR(18,15,30,5):: GO
TO 2500
```

```
2460 IF K>96 OR K<33 THEN CA
LL SOUND(100,300,3,-7,3):: G
OTO 2410 ELSE 2470
2470 CALL HCHAR(Y(K-32),X(K-
32)+2,K)
2480 FOR I=1 TO 30 :: NEXT I
 :: CALL HCHAR(Y(K-32),X(K-3
2)+2,32)
2490 GOTO 2410
2500 CALL HCHAR(18,15,32,5):
: GOTO 2410
2510 DISPLAY AT(24,3):"PRESS
 A KEY TO ERASE KEYS"
2520 CALL KEY(0,K,S):: IF S=
0 THEN 2520 :: DISPLAY AT(24
,3):RPT$("",25)
2530 FOR I=12 TO 59 :: CALL
HCHAR(Y(I),X(I)+2,32):: NEXT
 I :: RETURN
2540 SUBEND
2550 !@P-
2560 END
```

DER TI ALS SCHREIB-MASCHINEN LEHRER

Fortsetzung von Seite 128

wie in allen anderen Unterprogrammen lediglich FCTN 9 drücken.

Wenn Sie 3 drücken, gelangen Sie in das Unterprogramm „Nachschreiben vorher eingegebener Texte". Nun können Sie einen vorher eingegebenen Text nachschreiben. Dazu erscheint zuerst einmal eine kurze Erklärung auf dem Bildschirm, was in diesem Unterprogramm gemacht werden muß.

Im oberen Teil des Bildschirms wird jeweils eine Zeile angezeigt, die Sie dann in der Zeile etwas darunter nachschreiben müssen. Auch hier müssen Sie wieder spätestens am Ende der Zeile ENTER drücken. Wenn Sie dies gemacht haben, wird die nächste Zeile eingeblendet, die Sie dann nachschreiben müssen. Haben Sie alle 10 Zeilen nachgeschrieben, müssen Sie dann etwas warten, weil der Rechner dann überprüft, ob Sie Fehler beim Abschreiben gemacht haben. Die Anzahl der Fehler wird dann angezeigt. Danach können Sie dann wieder wählen, ob Sie mit FCTN 9 ins Menue zurück wollen, ob Sie mit 2 den Text nochmal nachschreiben wollen oder ob Sie mit 1 den nachgeschriebenen und den nachzuschreibenden Text angezeigt haben wollen.

Wenn Sie 4 drücken, können Sie messen, wie schnell Ihre Schreibgeschwindigkeit ist. Zuerst erscheint auch hier wieder eine kurze Einleitung, was in diesem Unterprogramm zu machen ist. Wenn Sie sich die kurze Einleitung durchgelesen haben, müssen Sie eine Taste drücken, damit das Programm fortfährt.

Wenn Sie 5 im Hauptmenü gewählt haben, druckt der Computer Ihnen noch einmal die TI-Tastatur auf dem Bildschirm aus, damit Sie noch einmal einen Überblick bekommen, wo die einzelnen Tasten auf der Tastatur liegen. Das können Sie natürlich auch auf der richtigen Tastatur machen, aber erstens sollen Sie bei dem Programm nie auf die Tastatur sehen und zweitens gewöhnen Sie sich dann besser an die Bildschirmtastatur, wenn Sie z.B. mit Nr. 1 des Hauptmenüs arbeiten wollen.

Falls Sie 6 drücken, können Sie das Programm beenden. Zuerst werden Sie aber noch gefragt, ob Sie ein neues Programm laden wollen? Wenn Sie ja drücken, gibt der Rechner automatisch die Anweisungen auf dem Bildschirm aus, um ein neues Programm zu laden (z.B. Rewind Cassette Tape). Wenn Sie Besitzer einer Diskettenstation sind, müssen Sie nur die Zeile 780 ändern. Wenn Sie aber statt J, N drücken, wird das Programm beendet.

Variablenliste:
STRINGS:
JN$ — String für die JA/NEIN-Abfrage in Nummer 6 des Hauptmenüs
EI$ ( ) — String (Datenfeld) für den nachzuschreibenden Text in „Nachschreiben vorher eingegebener Texte"
NA$ ( ) — String (Datenfeld) für den nachgeschriebenen Text in „Nachschreiben vorher eingegebener Texte"

VARIABLE:
I, I1, I2 — Variable für FOR-NEXT Schleifen
ZAE — Variable für die Anzahl der Fehler in „Nachschreiben vorher eingegebener Texte"
TI — Variable für die gebrauchten Zeiteinheiten in „Schreiben auf Zeit von freien Texten"
SE — Variable für den senkrechten Standpunkt des Cursors in „Schreiben auf Zeit von freien Texten"
WA — Variable für den waagerechten Standpunkt des Cursors in „Schreiben auf Zeit von freien Texten"
K, S — Variable für die CALL KEY-Abfrage
DJ — Wert für den waagerechten Standpunkt der Tasten auf der Bildschirmtastatur
DD — Zufallswert für die Auswahl der Symbole in Nr. 2 des Hauptmenüs
X1, Y1 — Werte für die „Data-Auslesung"
M ( ) — Variable (Datenfeld) für den senkrechten Standpunkt der Tasten auf der Bildschirmtastatur
X ( ) — Variable (Datenfeld) für den X-Wert der einzelnen Symbole auf der Bildschirmtastatur
Y ( ) — Variable (Datenfeld) für den Y-Wert der einzelnen Symbole auf der Bildschirmtastatur

Dirk Junghans

★

SPRITE-KONSTRUKTEUR

Wie hat man früher Sprites auf dem TI-99/4A erstellt? Man nahm Zettel und Bleistift und zeichnete zunächst einen Raster mit 16 mal 16 Feldern auf das Papier. Dann mußte man sich zunächst das Sprite gründlich überlegen, denn probieren war viel zuviel Arbeit. Schließlich kritzelte man das Sprite in den Raster und wenn es dann doch nicht den Anforderungen entsprach, blieb einem nichts anderes übrig als der Griff zum Radiergummi. Der Fehler wurde also nun ausradiert, die Rasterlinien nachgezogen und die Prozedur von neuem wiederholt.

Dann kam der peinlichste Teil der Arbeit: Mühevoll mußte man in Zusammenarbeit mit dem Bedienungsbuch die Hexcodes raussuchen — dauerndes Vergleichen mit der Zeichnung und dem Anleitungsbuch. Hatte man nun schließlich tatsächlich die sechzehn richtigen Codes herausgeschrieben, so sah man sich zunächst einmal das neue Sprite am Bildschirm an. Dazu mußte die Kolonne natürlich zuerst eingegeben werden, ein eigenes Programm dafür geschrieben werden, usw.

Jeder TI-99-Besitzer kennt diese Arbeiten nur zu gut. Oftmals verdirbt nur der Gedanke an sie dem Programmierer die Lust, ein neues Programm zu machen.

Wie kann man nun Sprites auf dem TI-99/4A erstellen? Ganz einfach: Mit dem Spriteconstruction-Set. Legen Sie Ihre Zettel, Bleistifte, Radiergummi und Bedienungsanleitungen ruhig zur Seite. Mit dem Spriteconstruction-Set wird das Erstellen von Sprites ein Kinderspiel. Sie malen die Form des Sprites einfach auf den Bildschirm, lassen es vom Computer in jeder beliebigen Darstellungsart bearbeiten, können sich das Ganze auch in „Natura" anschauen und bei Mißfallen nach Belieben Änderungen vornehmen. Auch die lästige Arbeit der Hex-Code-Bestimmung übernimmt für Sie nun der Computer. Und als Krönung haben Sie die Möglichkeit, bis zu 18 Sprites „aufzuheben", um sie später nochmals anzusehen, oder mit anderen Sprites zu vergleichen.

Spriteconstruction-Set

Ein extrem leistungsfähiges Hilfsprogramm. Mit dem Sprite-Construction Set können Sprites in Sekundenschnelle erstellt werden. Was bisher bie der Entwicklung von eigenen Progammen die anstrengendste und gefürchtetste Arbeit war, wird nun zur unterhaltsamen Beschäftigung. Ohne Bleistift, Zettel und Radiergummi malen Sie Ihre Sprites einfach auf den Bildschirm, lassen sie vom Computer drehen oder wenden, sehen sich Ihre Meisterwerke in Natura an und lassen auch den Hexcode vom Computer berechnen.

Die Möglichkeiten des Programms sind so umfangreich, daß sie hier gar nicht alle angeführt werden können. Eine sechsseitige Programmdokumentation liegt diesem Schreiben bei. Jeder TI-User wird das Sprite-Constructionset sicher sehr bald zu schätzen wissen!

Nach dem Starten des Programms mit RUN bildet der Computer zunächst das Zeichenfeld am Bildschirm ab und gleich danach das Hauptmenue. Neben der ersten Auswahlmöglichkeit des Hauptmenues befindet sich ein weißer Pfeil.

Sie können diesen Pfeil bewegen, indem Sie den Joystick in die entsprechende Richtung drükken. Der Pfeil wesit dann auf den nächsten Namen der Liste.

Beachten Sie die anderen Namen der Liste zunächst nicht, und führen Sie den Pfeil neben das Wort „WORKING".

Wenn Sie nun den Actionknopf Ihres Joysticks drücken, gelangen Sie in das Arbeitssystem, in dem Sie Ihre Sprites erstellen können. Die anderen Teile des Hauptmenues werden für Sie erst interessant, wenn Sie bereits erstellte Sprites verwalten (Speichern, aufrufen) wollen.

Im Arbeitsmodus wird zunächst das Arbeitsmenue aufgelistet. Es besteht aus folgenden Auswahlmöglichkeiten:

SYMMETRY, INVERT, MIRROR, ROTATION
HEX IN, HEX OUT, LOOK und READY

Beachten Sie vorläufig diese Auswahlmöglichkeiten noch nicht und drükken Sie den Joystick nach rechts.

Sie werden bemerkt haben, daß der weiße Pfeil neben dem Wort SYMMETRY des Menues verschwunden ist. Das bebedeutet, daß Sie das Menue verlassen haben, und sich jetzt im Arbeitsmodus befinden.

Im Zeichenfeld befindet sich nun ein Zielkreuz, das Sie in alle Richtungen bewegen können. Wenn Sie den Actioknopf drücken, können Sie nun Punkte Feld fahren, auf dem sich bereits ein Punkt befindet und Sie abermals den Actionknopf drücken, wird er wieder gelöscht.

Auf diese Art und Weise können Sie nun völlig problemlos Ihr Sprite „malen". Wenn Sie längere Linien ziehen müssen, oder Flächen ausmalen wollen, dann müssen Sie bei gedrücktem Feuerknopf in die entsprechende Richtung drücken und schon wird eine Linie gezogen. Jede Aktion Ihrerseits wird mit einem akustischen Signal untermalt.

Sie können den Arbeitsmodus wieder verlassen, indem Sie mit dem Zielkreuz über den Zeichenfeldrand hinaussteuern. Sobald dies geschehen ist, verschwindet das Zielkreuz und Sie befinden sich wieder im Auswahlmodus des Menues.

Das Arbeitsmenue

Das Arbeitsmenue ist imstande, Sie bei der Erstellung Ihrer Sprites wesentlich zu unterstützen. Sie können ein Programm des Arbeitsmenues auf-

rufen, indem Sie den Pfeil neben das gewünschte Wort steuern und dann den Actionknopf drücken. Ein weißer Punkt neben dem Wort zeigt Ihnen dann an, daß das Programm aktiviert wurde.

Symmetry:

Das Symmetry-Programm hilft Ihnen, wenn sie symmetrische Figuren erstellen wollen. Sie müssen dann, je nach Art der Symmetrie, nur ein Viertel oder die Hälfte der Zeichenfläche „bemalen" und lassen dann den Computer für Sie weitermalen.

Wenn Sie im Arbeitsmenue SYMMETRY gewählt haben, gelangen Sie zunächst in ein kleines Nebenmenue.

Hier stehen H-AXIAL, V-AXIAL und CIRCULAR zur Auswahl.

Neben den Wörtern ist ein Symbol abgebildet, welches veranschaulicht, welches Viertel bzw. welche Hälfte Sie bemalen müssen.

Circular:

Wenn Sie dieses Programm wählen, setzt der Computer Zirkularsymmetrie voraus. Das linke obere Veirtel wird dann zirkularsymmetrisch vervielfacht.

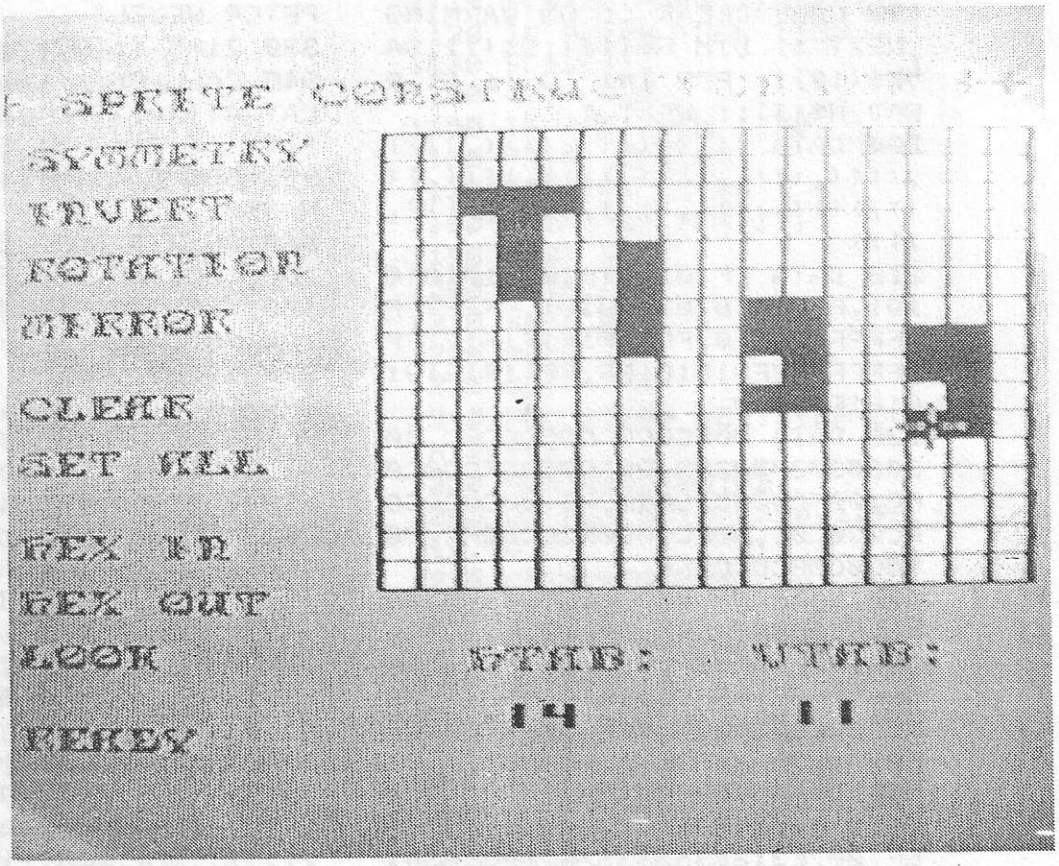
Dieses Programm eignet sich ideal zur Gestaltung von Spielbällen oder ähnlichen symmetrischen Figuren. Sie müssen lediglich ein Viertel des Balles zeichnen, steigen dann in die Programme SYMMETRY, CIRCULAR ein, und der Computer zeichnet das Gebilde fertig.

H-Axial:

Das Programm H-Axial setzt die horizontale Mittelachse als Symmetrieachse voraus. Zeichnen müssen Sie, bevor Sie dieses Programm wählen, daher die obere Hälfte des Sprites. Die untere Hälfte zeichnet dann der Computer.

V-Axial:

Das Programm V-Axial setzt die vertikale Mittelachse als Symmetrieachse voraus. Hier muß die linke Hälfte des Sprites gezeichnet sein, damit der Computer die andere Hälfte erstellen kann.



Invert:

Das INVERT-Programm invertiert ein fertig gezeichnetes Sprite. Das heißt, alle schwarzen Punkte im Zeichenfeld werden weiß und weiße Punkte werden schwarz. Wie bei allen anderen Hilfsprogrammen des Arbeitsmenues können Sie nach Verwendung des Programms wieder in den Arbeitsmodus einsteigen und Veränderungen vornehmen.

Mirror:

Das Mirror Programm spiegelt ein bereits fertiges Sprite. Auch hier gelangen Sie in ein Nebenmenue, welches Sie fragt, ob HORIZONTAL oder VERTIKAL gespiegelt werden soll.

Horizontal:

Spiegelt Horizontal. Ideal bei der Erstellung von Spielfiguren! Sie zeichnen zum Beispiel nur die Figur, die nach links geht, spiegeln diese dann horizontal und schon haben Sie die rechte Figur.

Vertikal:

Genauso wie HORIZONTAL. Es wird allerdings vertikal gespiegelt.

Rotation:

Wenn Sie das Rotation-Programm gewählt haben, taucht unterhalb des Zeichenfeldes ein neues Menue auf, welches sich erkundigt, ob Sie das Sprite nach links oder nach rechts rotieren lassen wollen.

Wenn Sie nun links gewählt haben, führt das Sprite eine 90 Grad-Dehnung nach links aus. Wählen Sie rechts, so geschieht derselbe Vorgang nach rechts.

Durch öftere Anwendung dieses Programmes kann man ein Sprite in jeder beliebigen Richtung darstellen. Bei der Definition von Spielfiguren ist dieses Programm unentbehrlich.

So wie die vorangegangenen vier Programme ist auch das nun folgende fünfte Programm des Arbeitsmenues bei der Erstellung von Sprites behilflich:

Bitte lesen Sie weiter auf Seite 139

```
10 ! **********************
11 ! *                    *
12 ! * SPRITE-CONSTRUCTION*
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *    Peter Wenzel    *
17 ! *                    *
18 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     9130 Bytes     *
28 ! *                    *
29 ! **********************
30 !
40 ! Bei diesem Programm hab
en wir auf die Hilfspunkte v
erzichtet ! (Die Redaktion)
190 CALL CLEAR :: ON WARNING
 NEXT :: DIM H$(16),D$(1),SA
VE$(18):: FOR I=1 TO 16 :: R
EAD H$(I):: NEXT I
200 DATA ((((,(((),(()(,(())
,()((,()(),())(,()))),)(((,)(
(),)()(,)()),))((,))(),))))(,
))))
210 DATA FF,01,0101010101010
101,FF0101010101010101,FFFFFFF
FFFFFFFFF,0,FFF1F1F1F1F1F1FF
,FFFFFFFF818181FF,FFF1F1F181
8181FF,0,0
220 DATA 00FCCCCCCCCCCCFC,00
303030303030303,00FC0C0CFCC0C0
FC,00FC0C0C7C0C0CFC,00CCCCCC
FC0C0C0C,00FCC0C0FC0C0CFC,00
C0C0C0FCCCCCFC
230 DATA 00FC0C0C0C0C0C0C,00
FCCCCCFCCCCCFC,00FCCCCCFC0C0
C0C,00FCCCCCFCCCCCCC,00F0CCC
CF0CCCCF,00FCC0C0C0C0C0FC,00
F0CCCCCCCCCCF,00FCC0C0F8C0C0
FC
240 DATA 00FCC0C0F8C0C0C,3C4
299A1A199423C,003F666C746466
EF,007E313B3633337E,003E6341
40603F1E,007E321119113F7E,00
7F33303C38317F,007F723239383
078
250 DATA 003E6648504F663E,00
7C603F3132327A03,007C38181C1
A183C2,003F13060606263F,007B
72363C363673,00783030307263
7F
260 DATA 007F2B2A49494963,00
7C76323A323477,003E634559617
F3E,007E33313F3E3078,003E634
559617F3E07,007F313A34323179
,003E413C06634F3F
270 DATA 007F492A08080810,00
6F2626666E7E37,0073622232321
E0C,0077222241494963,0079321
40814264F,0073416234181830,0
07F43260C1A317F,101010EE1010
1
280 DATA 0000181800001818,00
0080C7E7F7E0C08
290 FOR I=37 TO 93 :: READ I
$ :: CALL CHAR(I,I$):: NEXT
I
300 CALL COLOR(1,2,1,2,2,16,
3,5,1,4,5,1,5,9,1,6,9,1,7,9,
1,8,9,1,9,3,1,10,16,1,12,2,1
6)
310 CALL CLEAR :: CALL SCREE
N(8):: DISPLAY AT(1,1):"[[ S
PRITE CONSTRUCTIONSET [[" ::
 CALL SET
320 RESTORE 1240 :: CALL DIR
 :: DISPLAY AT(20,13):"@ BY
PETER WENZL"
330 Z1=5 :: S1=3 :: GOTO 350
340 CALL S :: CALL C :: DISP
LAY AT(20,13):" [[[[ READY [
[[[" :: Z1=Z :: S1=S
350 GOSUB 1160 :: IF X=4 THE
N 350
360 ON Z-4 GOTO 500,100,450,
100,100,100,100,370,100,390,
100,100,100,100,400,100,100,
100,100,100
370 GOSUB 930
380 CALL KEY(1,K,S):: IF K=1
8 THEN Z=12 :: S=3 :: GOTO 3
40 ELSE 380
390 GOSUB 1000 :: Z=14 :: GO
TO 340
400 IF HCV=0 THEN CALL SOUND
(3000,440,0):: DISPLAY AT(20
,13):"CAN NOT BE SAVED" :: D
ISPLAY AT(22,13):"USE HEXOUT
 BEVOR" :: CALL SOUND(1,440,
0):: Z=19 :: S=3 :: GOTO 340
410 IF SN>17 THEN CALL SOUND
(3000,440,0):: DISPLAY AT(20
,13):"SORRY\" :: DISPLAY AT(
22,13):"MEMORY FULL" :: CALL
 SOUND(1,440,0):: Z=19 :: S=
3 :: GOTO 340
420 SN=SN+1 :: CALL C :: CAL
L S :: DISPLAY AT(20,13):"SP
RITENAME\" :: ACCEPT AT(22,1
3)VALIDATE(UALPHA):I$ :: CAL
L S2 :: SAVE$(SN)=HEX$&HR$&I
$
430 CALL S :: DISPLAY AT(24,
13):"NUMBER";SN;"SAVED"
440 CALL KEY(1,K,Q):: IF K=1
8 THEN Z=19 :: S=3 :: GOTO 3
40 ELSE 440
```

```
450 IF ON=0 THEN Z=7 :: S=3
:: GOTO 340 :: ELSE DISPLAY
AT(20,13):"SPRITENUMBER\" ::
 CALL S :: ACCEPT AT(20,27):
Q :: IF Q<1 OR Q>SN THEN 450
460 CALL S2 :: DISPLAY AT(22
,13):"SPRITENAME\" :: DISPLA
Y AT(24,13):SEG$(SAVE$(Q),12
9,16):: HEX$=SEG$(SAVE$(Q),1
,64):: HR$=SEG$(SAVE$(Q),65,
64)
470 HCV=1 :: GOSUB 890 :: CA
LL S2 :: CALL S2 :: CALL S2
480 CALL KEY(1,K,Q):: IF Q=0
 THEN 480 ELSE CALL S :: Z=7
 :: S=3 :: GOTO 340
490 CALL S :: CALL C :: DISP
LAY AT(20,13):" [[[[ READY [
[[[" :: Z1=Z :: S1=S :: GOTO
 520
500 RESTORE 1250 :: CALL DIR
510 CALL C :: CALL DELSPRITE
(ALL):: Z1,S1=3
520 GOSUB 1160 :: IF X=4 THE
N 1100
530 IF Z=24 THEN 320
540 ON Z-2 GOTO 550,100,620,
100,650,100,720,100,100,770,
100,790,100,100,790,100,910,
100,990
550 HCV=0 :: CALL SOUND(200,
880,9,770,9,660,9):: DISPLAY
 AT(20,14):"' AXIAL\ H ] ,"
:: DISPLAY AT(22,14):"' AXIA
L\ V ] +" :: DISPLAY AT(24,1
4):"' CIRCULAR ] -"
560 Z1=20 :: S1=16 :: GOSUB
1160 :: IF X=4 THEN 560
570 IF Z=20 THEN 600 ELSE IF
 Z=22 THEN 610
580 FOR I=3 TO 10 :: CALL SO
UND(-4250,-1,9):: FOR Q=15 T
O 22
590 CALL GCHAR(I,Q,C1):: CAL
L HCHAR(I,45-Q,C1):: CALL HC
HAR(21-I,45-Q,C1):: CALL HCH
AR(21-I,Q,C1):: NEXT Q :: NE
XT I :: GOTO 510
600 FOR I=3 TO 10 :: CALL SO
UND(-4250,-2,9):: FOR Q=15 T
O 30 :: CALL GCHAR(I,Q,C1)::
 CALL HCHAR(21-I,Q,C1):: NEX
T Q :: NEXT I :: GOTO 510
610 FOR I=3 TO 18 :: CALL SO
UND(-4250,-3,0):: FOR Q=15 T
O 22 :: CALL GCHAR(I,Q,C1)::
 CALL HCHAR(I,45-Q,C1):: NEX
T Q :: NEXT I :: GOTO 510
620 HCV=0 :: DISPLAY AT(20,1
3):" [[ INVERTING [[" :: FOR
 I=3 TO 18 :: CALL SOUND(-20
00,-4,9):: FOR Q=15 TO 30 ::
 CALL GCHAR(I,Q,K)
630 IF K=40 THEN CALL HCHAR(
I,Q,41)ELSE CALL HCHAR(I,Q,4
0)
640 NEXT Q :: NEXT I :: GOTO
 490
650 HCV=0 :: D$(0),D$(1)=""
:: CALL SOUND(200,880,9,770,
9,660,9):: DISPLAY AT(22,13)
:"' LEFT" :: DISPLAY AT(24,1
3):"' RIGHT" :: Z1=22 :: S1=
15
660 GOSUB 1160 :: IF X=4 THE
N 660 ELSE IF Z=22 THEN 700
670 DISPLAY AT(20,13):"[[[ R
OTATION [[["
680 FOR A=0 TO 1 :: FOR I=15
 TO 22 :: CALL SOUND(-4000,-
7,9):: FOR Q=18 TO 3 STEP -1
 :: CALL GCHAR(Q,I+A*8,C1)::
 D$(A)=D$(A)&CHR$(C1):: NEXT
 Q :: NEXT I :: NEXT A
690 GOSUB 1260 :: Z=7 :: S=3
 :: GOTO 490
700 DISPLAY AT(20,13):"[[[ R
OTATION [[["
710 FOR A=0 TO 1 :: FOR I=30
 TO 23 STEP -1 :: CALL SOUND
(-4000,-6,9):: FOR Q=3 TO 18
 :: CALL GCHAR(Q,I-A*8,C1)::
 D$(A)=D$(A)&CHR$(C1):: NEXT
 Q :: NEXT I :: NEXT A :: GO
TO 690
720 CALL CHAR(120,"0024466FFF
F6624001183C7E18187E3C18")::
HCV=0 :: CALL SOUND(200,880,
9,770,9,660,9):: DISPLAY AT(
22,13):"' HORIZONTAL ] x"
730 DISPLAY AT(24,13):"' VER
TICAL   ] y" :: Z1=22 :: S1=
15 :: GOSUB 1160 :: DISPLAY
AT(20,13):"[[[[ MIRROR [[[["
 :: IF Z=22 THEN 760
740 FOR I=15 TO 30 :: CALL S
OUND(-4000,-3,9):: FOR Q=1 T
O 8 :: CALL GCHAR(Q+2,I,C1):
: CALL GCHAR(19-Q,I,C2):: CA
LL HCHAR(Q+2,I,C2):: CALL HC
HAR(19-Q,I,C1):: NEXT Q :: N
EXT I
750 CALL S :: CALL C :: Z=9
:: S=3 :: GOTO 490
760 FOR I=3 TO 18 :: CALL SO
UND(-4000,-2,9):: FOR Q=1 TO
 8 :: CALL GCHAR(I,Q+14,C1):
: CALL GCHAR(I,31-Q,C2):: CA
LL HCHAR(I,Q+14,C2):: CALL H
CHAR(I,31-Q,C1):: NEXT Q ::
NEXT I :: GOTO 750
770 HCV=0 :: CALL SET :: GOT
```

```
0 490
780 HCV=0 :: FOR I=3 TO 18 :
: CALL SOUND(-10,-6,0):: CAL
L HCHAR(I,15,41,16):: NEXT I
 :: GOTO 490
790 CALL CHAR(120,"FFF1F1F18
18181FFFF818181F1F1F1FFFF8F8
F8F818181FFFF8181818F8F8FFF"
):: CALL S :: DISPLAY AT(20,
13):"INSERT HEXCODE\" :: HR$
,HEX$=""
800 HCV=1 :: FOR I=1 TO 4 ::
 DISPLAY AT(22,13):"PARTNUMB
ER";I;"] ";CHR$(119+I)
810 S=15
820 CALL HCHAR(24,S,105)
830 CALL KEY(3,K,Q):: IF Q=0
 THEN 830 ELSE IF K<48 OR K>
57 THEN IF K<65 OR K>70 THEN
 CALL SOUND(100,-1,0):: GOTO
 830
840 CALL SOUND(12,999,8,-7,8
):: IF K>57 THEN K=K-7
850 CALL HCHAR(24,S,K):: HEX
$=HEX$&CHR$(K):: S=S+1 :: IF
 K>57 THEN HR$=HR$&CHR$(K+7)
ELSE HR$=HR$&CHR$(K)
860 IF S<31 THEN 820 ELSE CA
LL SOUND(50,-5,0):: CALL SOU
ND(50,-6,0):: CALL SOUND(50,
-3,0):: CALL HCHAR(24,15,32,
17)
870 NEXT I
880 GOSUB 890 :: Z=17 :: S=3
 :: GOTO 490
890 K=1 :: FOR I=13 TO 21 ST
EP 8 :: FOR Q=3 TO 18 :: DIS
PLAY AT(Q,I)SIZE(4):H$(ASC(S
EG$(HEX$,K,1))-47):: DISPLAY
 AT(Q,I+4)SIZE(4):H$(ASC(SEG
$(HEX$,K+1,1))-47)
900 K=K+2 :: NEXT Q :: NEXT
I :: RETURN
910 GOSUB 930
920 CALL KEY(1,K,S):: IF K=1
8 THEN Z=19 :: S=3 :: GOTO 4
90 ELSE 920
930 CALL HCHAR(20,15,32,16):
: Z1=21 :: S1=14 :: HEX$,HR$
="" :: HCV=1
940 FOR SF=0 TO 8 STEP 8 ::
FOR ZF=0 TO 8 STEP 8 :: FOR
A=0 TO 7 :: FOR Q=0 TO 4 STE
P 4 :: C2=0 :: FOR I=0 TO 3
:: CALL GCHAR(3+ZF+A,18+SF-I
+Q,C1)
950 C2=C2+2^I*(C1-48):: NEXT
 I :: S1=S1+1 :: HEX$=HEX$&C
HR$(48+C2)
960 IF C2>9 THEN HR$=HR$&CHR
$(C2+55)ELSE HR$=HR$&CHR$(C2
+48)
970 IF S1>30 THEN S1=15 :: Z
1=Z1+1
980 CALL SOUND(10,-5,0):: CA
LL HCHAR(Z1,S1,48+C2):: NEXT
 Q :: NEXT A :: NEXT ZF :: N
EXT SF :: CALL S2 :: CALL S2
 :: CALL S2 :: RETURN
990 GOSUB 1000 :: Z=21 :: GO
TO 490
1000 IF HCV=0 THEN CALL SOUN
D(3000,440,0):: CALL C :: DI
SPLAY AT(20,13):"WARNING\" :
: DISPLAY AT(22,13):"USE HEX
OUT BEVOR" :: GOTO 1090
1010 CALL CHAR(128,HR$):: CA
LL C :: CALL SOUND(200,880,9
,770,9,660,9):: DISPLAY AT(2
0,13):"' COLOR" :: DISPLAY A
T(22,13):"' MAGNIFY" :: DISP
LAY AT(24,13):"' END"
1020 Q=3 :: CALL MAGNIFY(3):
: CALL SPRITE(#25,128,2,160,
208)
1030 Z1=20 :: S1=15 :: GOSUB
 1160 :: IF X=4 THEN 1030
1040 IF Z=24 THEN 1090
1050 IF Z=22 THEN 1070 ELSE
ACCEPT AT(20,21)SIZE(2)VALID
ATE(DIGIT)BEEP:I$ :: IF I$="
" THEN 1050 ELSE CO=VAL(I$)
1060 IF CO<1 OR CO>16 THEN 1
050 ELSE CALL COLOR(#25,CO):
: GOTO 1030
1070 IF Q=3 THEN CALL MAGNIF
Y(4):: Q=4 ELSE CALL MAGNIFY
(3):: Q=3
1080 CALL HCHAR(22,15,96)::
GOTO 1030
1090 CALL DELSPRITE(ALL):: C
ALL SOUND(1,440,0):: S=3 ::
RETURN
1100 CALL C :: DISPLAY AT(21
,15):"HTAB\ VTAB\" :: HCV=0
 :: CALL MAGNIFY(2):: Z,S=1
:: CALL SPRITE(#9,91,14,7+Z*
8,102+S*8):: DISPLAY AT(23,1
7):"1       1"
1110 CALL KEY(1,K,I):: IF K<
>18 THEN 1130 ELSE CALL GCHA
R(Z+2,S+14,I)
1120 IF I=40 THEN CALL SOUND
(100,220,0,247,0,262,0):: CA
LL HCHAR(Z+2,S+14,41)ELSE CA
LL SOUND(99,440,0,494,0,524,
0):: CALL HCHAR(Z+2,S+14,40)
1130 CALL JOYST(1,X,Y):: IF
X=0 AND Y=0 THEN 1110
1140 Z=Z-Y/4 :: S=S+X/4 :: I
F Z>16 OR Z<1 THEN 510 ELSE
IF S>16 OR S<1 THEN 510
```

```
1150 CALL SOUND(30,-2,0):: C
ALL LOCATE(#9,7+Z*8,102+S*8)
:: DISPLAY AT(23,15):USING "
###    ###":S,Z :: GOTO 1110
1160 Z=Z1 :: S=S1 :: CALL S
:: CALL HCHAR(Z,S,104)
1170 CALL KEY(1,K,I):: IF K=
18 THEN CALL HCHAR(Z,S,105):
: :: CALL S2 :: GOTO 1230
1180 CALL JOYST(1,X,Y):: IF
X=4 THEN 1220
1190 IF Y=0 THEN 1170 ELSE C
ALL HCHAR(Z,S,96)
1200 Z=Z-Y/4 :: IF Z>24 OR Z
<3 THEN 1160
1210 CALL GCHAR(Z,S,FELD)::
IF FELD<>96 THEN 1200 ELSE C
ALL SOUND(100,-1,9):: CALL H
CHAR(Z,S,104):: GOTO 1170
1220 CALL S :: CALL HCHAR(Z,
S,96)
1230 RETURN
1240 DATA CREATE\,,' WORKING
,,' OLD ONE,,,OUTPUT\,,' HEX
CODE,,' LOOK,,,STORAGE\,,' S
AVE,,,,,,
1250 DATA ' SYMMETRY,,' INVE
RT,,' ROTATION,,' MIRROR,,,'
 CLEAR,,' SET ALL,,,' HEX IN
,,' HEX OUT,,' LOOK,,,' READ
Y,
1260 FOR A=0 TO 1 :: FOR I=0
 TO 7 :: DISPLAY AT(I+3+A*8,
13):SEG$(D$(A),I*16+1,16)::
NEXT I :: NEXT A :: RETURN
1270 SUB SET :: CALL SOUND(-
999,-5,9):: CALL VCHAR(3,14,
39,16):: CALL HCHAR(19,14,38
):: CALL HCHAR(19,15,37,16)
1280 FOR I=3 TO 18 :: CALL S
OUND(-10,-5,0):: CALL HCHAR(
I,15,40,16):: NEXT I :: CALL
 CHAR(96,"00552A552A552A55",
104,"0008080C7E7F7E0C08",105,"
0000103C3E3E3E1C"):: SUBEND
1290 SUB C :: FOR I=20 TO 24
 :: DISPLAY AT(I,13):"" :: N
EXT I :: SUBEND
1300 SUB S2 :: FOR I=0 TO 5
:: CALL SOUND(-200,110*2^I,4
):: NEXT I :: SUBEND
1310 SUB DIR :: FOR I=3 TO 2
4 :: DISPLAY AT(I,1)SIZE(11)
:"" :: NEXT I :: FOR I=3 TO
24 :: READ I$ :: IF I$="" TH
EN CALL SOUND(20,-3,9)ELSE C
ALL SOUND(20,-2,9)
1320 DISPLAY AT(I,1)SIZE(11)
:I$ :: NEXT I :: SUBEND
1330 SUB S :: CALL SOUND(-50
,330,4,220,4):: CALL SOUND(-
50,330,4,440,4):: SUBEND
1340 !
1350 ! *** PROGRAMMENDE ***
```

Fortsetzung von Seite 135

Hexin:

Hier haben Sie die Möglichkeit, den bereits existierenden Hexcode eines Sprites einzugeben. Ist dies geschehen, so können Sie an diesem Sprite nach Belieben Änderungen vornehmen, oder sich selbiges in „Natura" anschauen. (Dieses Programm mit dem Namen „LOOK" wird später erläutert!).

Dieses Programm ist sehr nützlich, wenn Sie sich die Grafiken anderer Listings einmal genauer ansehen wollen, bevor Sie das Listing eingeben, oder wenn Sie die Grafiken anderer Programme für eines Ihrer eigenen Produkte verwenden wollen.

Hex out:

Dieses Programm nimmt Ihnen die mühsamste Arbeit bei der Spritedefinierung ab. Es erzeugt den Hexcode des Sprites. Dieser wird unmittelbar unterhalb des Zeichenfeldes in vier Zeilen ausgegeben, wobei jede Zeile 16 Zeichen enthält und damit dem Hexcode eines Zeichens entspricht. (Ein Sprite besteht ja aus vier Zeichen!)

1. Zeile ..... linkes oberes Viertel des Sprites
2. Zeile ..... linkes, unteres Viertel des Sprites
3. Zeile ..... rechtes, oberes Viertel des Sprites
4. Zeile ..... rechtes, unteres Viertel des Sprites

Wenn Sie den Hexcode in einem Programm verwerten wollen, dann geben sie ihn einfach der Reihe nach von Zeile 1–4 ein.

Bevor Sie die Programme wie „SAVE" oder „LOOK" verwenden, müssen Sie das HEX OUT-Programm verwenden, da der Computer den Hex-Code auch für diese Programme zum Rechnen verwendet.

Look

Mit der Hilfe dieses Programmes können Sie sich Ihre fertigen Sprites in Originalgröße anschauen. Sobald Sie dieses Programm aufgerufen haben, erscheint das Sprite am rechten, unteren Bildschirmrand, und daneben eine neues Menue. Hier können Sie die Farbe des Sprites mittels COLOR ändern, oder es mittels MAGNIFY vergrößern. Eine neuerliche Anwendung von Magnify bringt das Sprite wieder in die Ursprungsgröße zurück.

Bevor Sie dieses Programm verwenden können, müssen Sie das Hex-Out Programm durchlaufen haben. Haben Sie dies vergessen, so weist der Computer darauf hin.

Wenn Sie genug gesehen haben, steigen Sie mittels END aus und kehren so wieder in das Arbeitsmenue zurück.

Zum Schluß die drei einfachsten Teile des Arbeitsmenues:

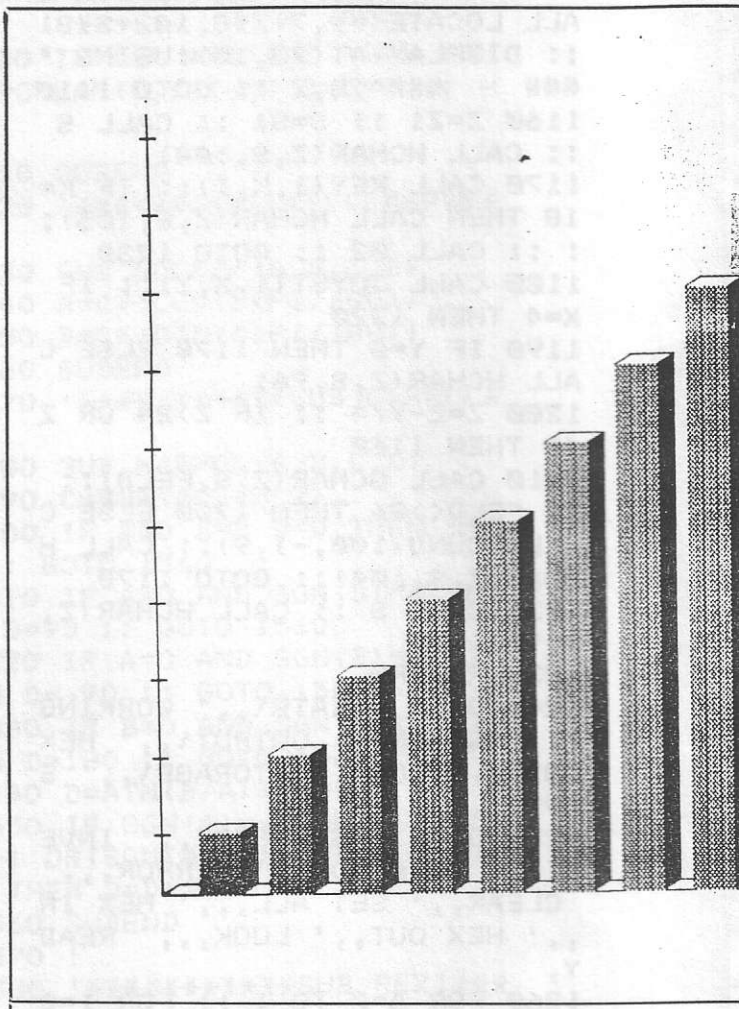
Clear

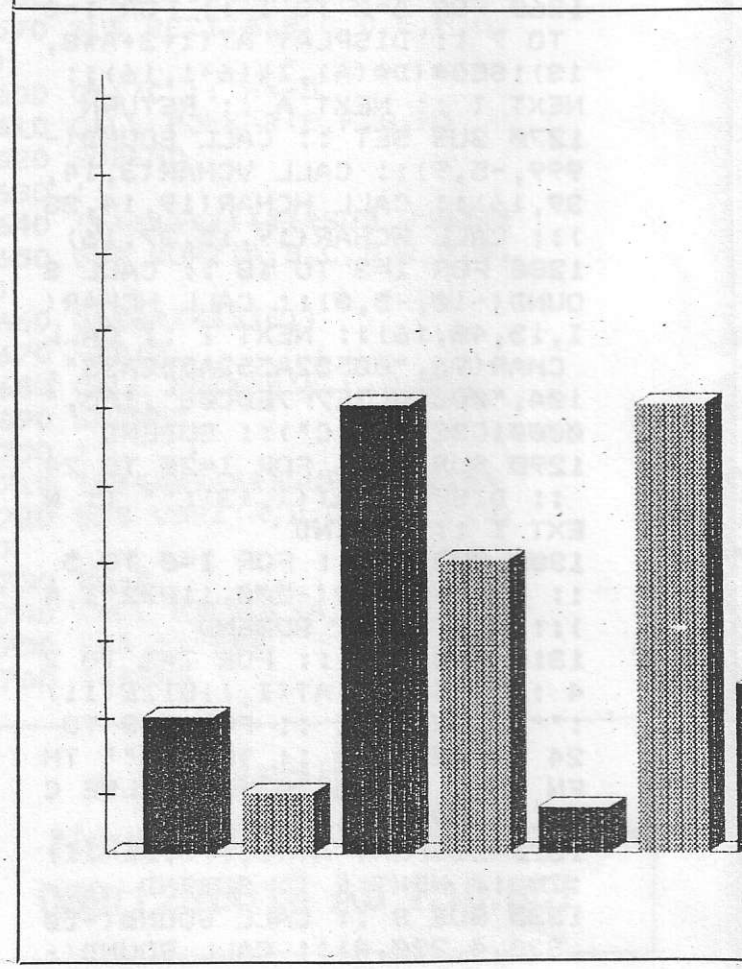
löscht das Zeichenfeld. Das dargestellte Sprite verschwindet.

Vorsicht: Sprites, die vorher gesaved wurden, sind nach der Anwendung von CLEAR nicht wieder aufrufbar. Sie sollten daher fertige Sprites, die Sie noch benötigen,

Bitte lesen Sie weiter auf Seite 146

WAS MAN SCHWARZ AUF WEISS BESITZT





Dies ist ein Programm, das dem creativen Anwender viele Variationsmöglichkeiten bietet, Säulen 3-dimensional mit dem Drucker RX-80 darzustellen.

Es ist für alle, die ihre Säulen schwarz auf weiß in der Hand halten wollen und auf größtmögliche Freiheit und Komfort bei der Gestaltung und Ausführung solcher Anwenderprogramme Wert legen, gedacht.

Dieses Programm ist in folgende Abschnitte unterteilt:

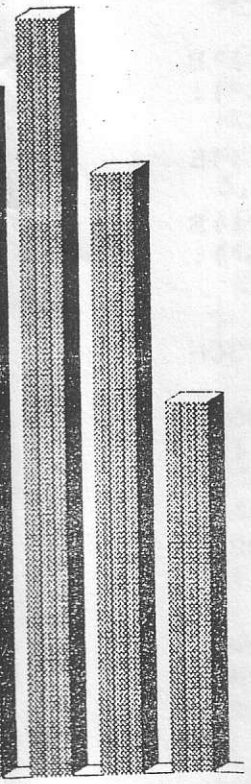
1. Eingabe: Hier werden alle wichtigen Vorausetzungen für den weiteren Ablauf von Ihnen eingegeben, wodurch die Variationen der verschiedenen Drucke erst zustande kommt. Sie können hier, im Dialog mit dem Computer, neben der Anzahl der Säulen auch entscheiden, ob die Y-Achse, ein Zwischenraum, oder im 2-Farben-Druck gedruckt werden soll.

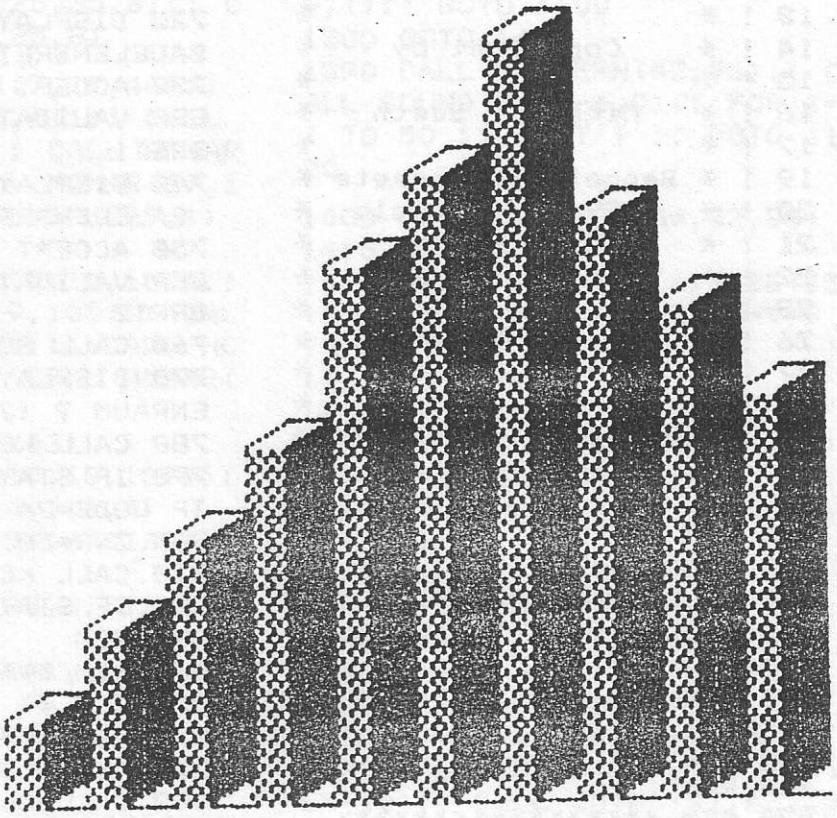
Zum 2-Farben-Druck ist anzumerken, daß hier in abwechselnder Folge 2 Säulen in verschiedenen Farben gedruckt werden, z.B. um einen Vergleich darzustellen. Eine weitere Möglichkeit, die Grafik ganz erheblich zu verändern, stellt die Eingabe der Säulenbreite dar.

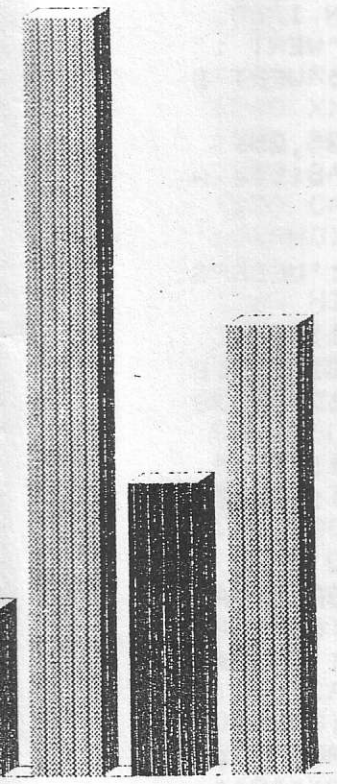
Mit der Eingabe des Faktors lassen Sie die Höhe der Säulen hochrechnen. Geben Sie z.B. den Faktor .5 ein, dann rechnet der Computer alle Werte auf 100 hoch.

2. Werteeingabe: Hier geben Sie die Werte für die Säulenhöhen ein. Angezeigt wird hier neben dem letzten von Ihnen eingegetippten Wert und der Anzahl der bisher eingetippten Säulen auch der höchste Wert, der nicht überschritten werden darf.

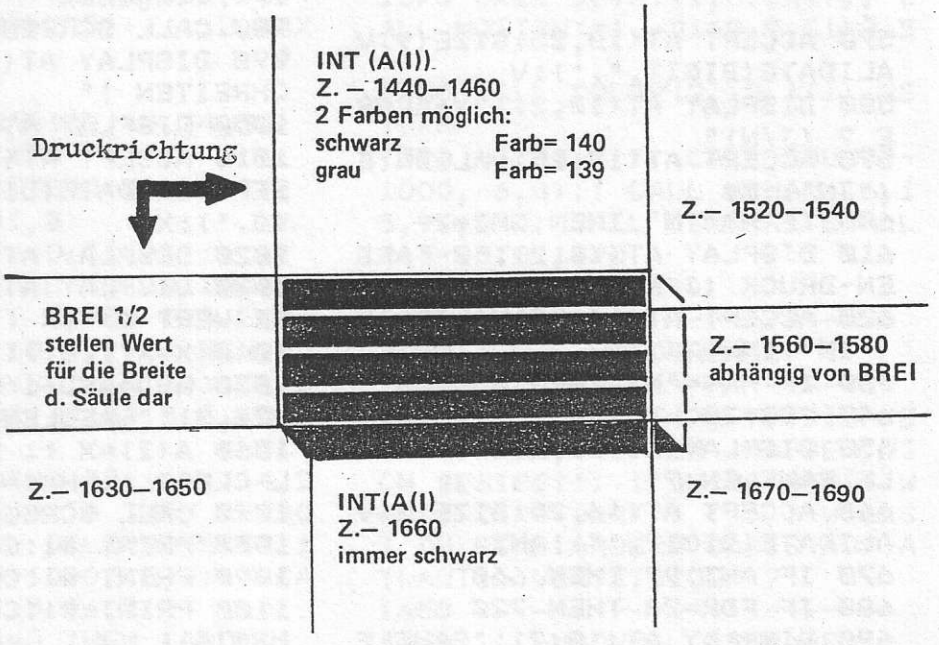
Michael Büth







Schematischer
Aufbau einer Säule
mit Verweis auf die Zeilennummern
wo die betreffenden Stücke entstehen
und gedruckt werden einschließlich
Kommentar



```
10 ! **********************
11 ! *                    *
12 ! *  SAEULENDIAGRAMM   *
13 ! *                    *
14 ! *   Copyright by     *
15 ! *                    *
16 ! *  Michael  Bueth    *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *    Epson  RX-80    *
23 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     4279 Bytes     *
28 ! *                    *
29 ! **********************

30 !
400 DIM A(99)
410 OPEN #1:"PIO"
420 CALL CHAR(135,"0000FF")
430 CALL CLEAR :: CALL SCREE
N(14)
440 FOR UI=1 TO 13 :: CALL C
OLOR(UI,16,2)
450 NEXT UI
460 REM
470 REM ******************
480 REM *  EINGABE DER   *
490 REM *                *
500 REM *VORAUSSETZUNGEN*
510 REM ******************
520 REM
530 DISPLAY AT(2,7):"SAUELEN
DIAGRAMM"
540 CALL HCHAR(3,9,135,15)
550 DISPLAY AT(5,6):"von mic
hael bueth "
560 DISPLAY AT(10,2):"FAKTOR
 :"
570 ACCEPT AT(10,25)SIZE(9)V
ALIDATE(DIGIT,"."):V
580 DISPLAY AT(12,2):"Y-ACHS
E ? (J/N)"
590 ACCEPT AT(12,25)VALIDATE
("JN"):M$
600 IF M$="N" THEN GHJ=2
610 DISPLAY AT(14,2):"2-FARB
EN-DRUCK (J/N)"
620 ACCEPT AT(14,25)VALIDATE
("JN"):HN$
630 IF HN$="N" THEN 650
640 FDR=20
650 DISPLAY AT(16,2):"WIEVIE
LE SAUELEN ?"
660 ACCEPT AT(16,25)SIZE(2)V
ALIDATE(DIGIT,""):ANZ
670 IF ANZ>99 THEN 660
680 IF FDR=20 THEN 720
690 DISPLAY AT(18,2):"SAUELE
NBREITE :"
700 ACCEPT AT(18,25)SIZE(2):
BREI2
710 GOTO 760
720 DISPLAY AT(18,2):"ERSTE
SAUELENBREITE ?"
730 ACCEPT AT(18,25)SIZE(1)B
EEP VALIDATE(DIGIT,"12345"):
BREI1
740 DISPLAY AT(20,2):"ZWEITE
 SAUELENBREITE ?"
750 ACCEPT AT(20,25)SIZE(1)B
EEP VALIDATE(DIGIT,"12345"):
BREI2
760 CALL SOUND(100,110,0)
770 DISPLAY AT(22,2):"ZWISCH
ENRAUM ? (J/N)"
780 CALL KEY(0,CODE,STATUS)
790 IF STATUS=0 THEN 780 ::
IF CODE=74 THEN 810
800 ZWR=50
810 CALL KEY(0,CODE,STATUS)
820 IF STATUS=0 THEN 810
830 REM
840 REM ******************
850 REM **  WERTEINGABE  **
860 REM ******************
870 REM
880 CALL CLEAR
890 DISPLAY AT(2,7):"SAUELEN
DIAGRAMM"
900 LI=50/V
910 CALL HCHAR(3,9,135,15)
920 FOR I=1 TO ANZ
930 IF ASD>ANZ-1 THEN 1760
940 DISPLAY AT(7,2):"WERT :"
950 DISPLAY AT(14,6):"WERT D
ARF NICHT ";LI
960 CALL HCHAR(8,4,135,25)
970 DISPLAY AT(9,8):"bitte w
ert eingeben "
980 CALL SCREEN(2)
990 DISPLAY AT(16,7):"UEBERS
CHREITEN !"
1000 DISPLAY AT(7,20):"? "
1010 ACCEPT AT(7,15)SIZE(5)B
EEP VALIDATE(DIGIT,"12345678
90."):X
1020 DISPLAY AT(7,15):"---"
1030 DISPLAY AT(22,3):"LETZT
ER WERT : ";X
1040 X=X/1.0 :: X=X*V
1050 ASD=ASD+1 :: DISPLAY AT
(20,3):"SAEULEN      : ";ASD
1060 A(I)=X :: NEXT I :: CAL
L CLEAR :: HAI=1
1070 CALL SCREEN(9):: ST=0
1080 PRINT #1:CHR$(27);"1";
1090 PRINT #1:CHR$(27);"E";
1100 PRINT #1:CHR$(27);"m";C
HR$(4);
1110 REM
```

```
1120 REM ********************
1130 REM **     AUSDRUCK     **
1140 REM ********************
1150 REM
1160 FOR I=1 TO ANZ
1170 GHJ=GHJ+1 :: IF GHJ>1 T
HEN 1330
1180 RESTORE 1950
1190 PRINT #1:CHR$(27);"L";C
HR$(6);CHR$(0);
1200 FOR TGH=1 TO 6
1210 READ BRED :: PRINT #1:C
HR$(BRED);:: NEXT TGH
1220 FOR AN=1 TO 10
1230 ST=ST+1
1240 IF ST<2 THEN 1260
1250 PRINT #1:CHR$(133);
1260 PRINT #1:CHR$(133);CHR$
(133);CHR$(133);CHR$(128);
1270 NEXT AN
1280 PRINT #1:CHR$(133)
1290 RESTORE 1960
1300 PRINT #1:CHR$(27);"L";C
HR$(9);CHR$(0);
1310 FOR RW=1 TO 9 :: READ D
FV :: PRINT #1:CHR$(DFV);
1320 NEXT RW :: PRINT #1:CHR
$(160):: GOSUB 1720
1330 PRINT #1:CHR$(27);"3";C
HR$(20);
1340 IF ASD>1 THEN 1350 :: G
OSUB 1720
1350 IF FDR=20 THEN 1370
1360 GOTO 1410
1370 REM
1380 ASD=ASD+1
1390 IF ASD=2*INT(ASD/2)THEN
 FARB=139 ELSE FARB=140
1400 GOTO 1420
1410 FARB=139
1420 BG=0
1430 IF A(I)<2 THEN GOSUB 17
20 :: BG=BG+1 :: GOTO 1590
1440 FOR DS=1 TO INT(A(I))-1
1450 PRINT #1:CHR$(FARB);
1460 NEXT DS
1470 REM
1510 BG=BG+1 :: IF BG>1 THEN
 1560
1520 RESTORE 1910 :: PRINT #
1:CHR$(27);"L";CHR$(8);CHR$(
0);
1530 FOR TG=1 TO 8 :: READ P
O :: PRINT #1:CHR$(PO);
1540 NEXT TG :: PRINT #1:CHR
$(160)
1550 GOTO 1590
1560 RESTORE 1920 :: PRINT #
1:CHR$(27);"L";CHR$(8);CHR$(
0);
1570 FOR HG=1 TO 8 :: READ H
U :: PRINT #1:CHR$(HU);
1580 NEXT HG :: PRINT #1:CHR
$(160)
1590 IF FARB=139 THEN 1610
1600 IF BG>BREI1-1 THEN 1620
 :: GOTO 1430
1610 IF BG>BREI2-1 THEN 1620
 :: GOTO 1430
1620 IF A(I)<2 THEN 1700
1630 RESTORE 1930 :: PRINT #
1:CHR$(27);"L";CHR$(12);CHR$
(0);
1640 FOR MN=1 TO 12 :: READ
LOK :: PRINT #1:CHR$(LOK);
1650 NEXT MN
1660 FOR DFE=1 TO A(I)-2 ::
PRINT #1:CHR$(140);:: NEXT D
FE
1670 RESTORE 1900 :: PRINT #
1:CHR$(27);"L";CHR$(8);CHR$(
0);
1680 FOR AF=1 TO 8 :: READ Q
 :: PRINT #1:CHR$(Q);
1690 NEXT AF :: PRINT #1:CHR
$(160)
1700 IF ZWR=50 THEN 1710 ::
GOSUB 1720
1710 NEXT I :: GOTO 1760
1720 RESTORE 1940 :: PRINT #
1:CHR$(27);"L";CHR$(9);CHR$(
0);
1730 FOR UJ=1 TO 9 :: READ X
CV :: PRINT #1:CHR$(XCV);
1740 NEXT UJ :: PRINT #1:CHR
$(160)
1750 RETURN
1760 FOR GH=1000 TO 110 STEP
 -50 :: CALL SOUND(-10,GH,7,
GH+20,20):: NEXT GH
1770 DISPLAY AT(10,1):"KOPIE
 ERWUENSCHT ? (J/N) "
1780 ACCEPT AT(10,26)SIZE(1)
VALIDATE("JN"):K$
1790 IF K$="N" THEN 1820 ELS
E 1800
1800 ST=0
1810 IF M$="N" THEN GHJ=2 EL
SE GHJ=0
1815 GOTO 1070
1820 DISPLAY AT(15,1):"NEUE
WERTEEINGABE UNTER"
1830 DISPLAY AT(17,1):"GLEIC
HEN VORRAUSSETZUNGEN ?"
1840 ACCEPT AT(21,16)SIZE(1)
VALIDATE("JN"):C$
1850 IF C$="J" THEN 1870
1860 GOTO 1880
1870 ASD=0 :: IF M$="N" THEN
 GHJ=2 ELSE GHJ=0
1875 GOTO 880
1880 CLOSE #1
```

```
1890 CALL SCREEN(2):: CALL C
LEAR :: END
1900 DATA 255,127,63,31,15,7
,3,255
1910 DATA 128,64,32,16,8,4,2
,1
1920 DATA 0,0,0,0,0,0,0,255
1930 DATA 255,128,192,224,24
0,248,252,254,255,255,255,25
5
1940 DATA 255,0,0,0,0,0,0,0,
255
1950 DATA 31,24,20,18,17,16
1960 DATA 255,0,0,0,0,64,32,
16,15
```

TIPS & TRICKS

DER KLEINE MUNTERMACHER

Eine kleine Hardwareerweiterung macht den langsamen TI 99/4A zum Rennwagen.

Haben Sie das auch schon einmal erlebt:
Sie haben eine ganz tolle Idee für ein Programm und bemerken beim Programmieren, daß Ihr Vorhaben ein Flop werden muß, da die notwendigen Routinen den Rechner viel zu langsam machen? Und dann ärgern Sie sich darüber, weil die Assemblerausrüstung für Ihren TI viel zu teuer ist? Das ist nun vorbei.
Wir haben für Sie eine Hardwareschaltung zur Assemblerprogrammierung getestet, die zum einen sehr günstig zu kaufen ist und zum anderen nur das Extended Basic-Modul benötigt.
Damit Sie uns richtig verstehen: den Cassettenrecorder brauchen Sie auch noch, denn irgendwo müssen Ihre Programme ja gespeichert werden. Sie haben richtig gelesen: Es wird kein Diskettenlaufwerk benötigt. Und jetzt in medias res.

DER KASTEN

Die kleine Hardwareschaltung findet in einem Kunststoffkasten von 100 mm x 50 mm x 25 mm Platz und besteht, wenn Sie das Gehäuse geöffnet haben, aus sehr wenigen Bausteinen. Auf der Platine finden sich zwei Widerstände, ein Chip vom Typ HM6116P-3, ein Speicherbaustein, und ein SN74LS138N. Ein kleiner Kippschalter wurde an der Gehäusestirnseite untergebracht. Der elektronische Aufbau birgt also weder Wunder noch technische Novation. Aber sinnreiche Details. Dieser kleine Kasten zeigt an beiden Längsseiten einen Peripheriebus. Mit dem einen Stecker können Sie damit Ihre Neuerwerbung an den Systembus Ihres TI anschließen und die andere Busseite für weitere Peripherie nutzen. Falls vorhanden. Die Stecker passen übrigens sauber an den TI-Bus, da klemmt oder wakkelt nichts. Ebenfalls von sehr guter Qualität ist das Kunststoffgehäuse. Es besteht aus zwei Hälften, die sehr stabil aussehen und mit vier starken Schrauben zusammengehalten werden. Einen Sturz von der Tischplatte dürfte das Gehäuse wohl aushalten. Aber versuchen Sie bitte nicht, mit Ihren Füßen die Stabilitätsgrenze des Gehäuses zu ergründen. Die beiden Gehäusehälften werden nicht einfach stumpf aufeinander gesetzt, sondern zeigen Nut und Feder. Das zeugt von durchdachter Konstruktion. Derart verbundene Bauteile können nicht wackeln oder zueinander verrutschen. Nur die Farbgebung der Gehäuseteile läßt zu wünschen übrig. Grau in Grau ist die Devise. Oberteil in Hellgrau Unterteil in Dunkelgrau. Aber Ihr TI soll ja auch kein buntes Kunstwerk sein.

AUSSTATTUNG

Das kleine Kästchen kommt natürlich nicht alleine. In seiner Begleitung finden sich eine Kassette und ein kleines Handbuch. Klein ist etwas untertrieben, denn im Format DIN A4 läßt es sich schwer verlieren und mit 25 Seiten hat es schon recht viel zu lesen.
Im Handbuch wird nicht nur beschrieben, wie Sie Ihr kleines Zauberkästchen anschließen müssen. Sie werden in einem Kurzlehrgang in die Assemblerkürzel eingewiesen. Und zwar in alle, die auf dem TMS 9900 lauffähig sind. Und das Ganze geschieht in deutscher Sprache. Obendrein findet sich ein kleines Beispielprogramm, das Ihnen den ersten Start leicht machen soll. Sie kommen aber in keinem Fall darum herum, sich selbst mit anderer Literatur zur Assemblerprogrammierung zu beschäftigen. Und da gibt es mittlerweile einiges auf dem Büchermarkt. Einen entsprechenden Vermerk mit Literaturhinweis finden Sie ebenfalls im Handbuch. Ein, wie wir meinen, lobenswerter Zug.
Die mitgelieferte Kassette hat dann endlich alles, was Sie sich schon lange gewünscht haben:
- Line by Line-Assembler
- Dissassembler
- Kassettenladeroutine
- Kassettenspeicherroutine

ANWENDUNG

Um Ihre Erweiterung in Betrieb nehmen zu können, genügt es, sie an den TI-Bus zu stecken. Dazu muß die Konsole ausgeschaltet sein. Der schon erwähnte Kippschalter soll in der linken Position stehen. Danach schalten Sie Ihr Gerät ein, rufen Ex-Basic auf und bringen den Kippschalter in die andere Position. Jetzt können Sie den Befehl CALL INIT eingeben. Nach Eingabe von NEW ist Ihr Rechner betriebsbereit. Wenn Sie jetzt den Line by Line-Assembler von Kassette laden und starten, können Sie sofort mit der Programmierung Ihrer Ideen beginnen. Auf Wunsch erfolgt ein Protokollausdruck über einen angeschlossenen Drucker. Der Assembler

ähnelt sehr stark dem vom Mini Mem, kann aber von der Geschwindigkeit nicht mithalten und hat leider auch einige Einschränkungen, die nicht verschwiegen werden:

- nur eine Zahl pro Data-Anweisung zulässig
- außer Indizierung keine Berechnung im Operandenfeld
- Programme sind auf eine bestimmte Adresse festgelegt
- keine vorgegebenen Definitionen
- das S-Symbol als Synonym für den Wert des Programmzählers ist nicht zulässig.

Mit diesen Einschränkungen läßt es sich aber ganz gut leben, zumal Sie mit diesem Assembler ein wirkungsvolles Instrument in der Hand haben, das es auch erlaubt, direkte Manipulationen im VDP vorzunehmen. Die gesamte Handhabung des Assemblers ist problemlos zu nennen und bereitet viel Spaß, wenn Sie sich erst einmal eingearbeitet haben.
Eine weitere wertvolle Hilfe stellt der Disassembler dar, mit dem Sie z.B.: fremde Maschinenprogramme untersuchen können.
Damit Ihre Programme nicht verloren gehen, können Sie alles auf Kassette abspeichern. Sie brauchen keine schwer verständlichen Befehle einzugeben, denn die Speicherroutine sucht sich die Namenstabelle, Programmstartadresse und Länge des Programms selber.
Genauso unkompliziert funktioniert das Wiedereinlesen Ihres Programms. Da kann wirklich nichts schief gehen. Dafür sorgt die Einleseroutine.

ERKLÄRUNGEN

Die auf Kassette mitgelieferten Programme befinden sich schon recht lange auf dem Markt. Daraus macht auch das Handbuch kein Geheimnis. Es handelt sich hierbei um Ex-Basic-Programme, die erstmalig 1983 veröffentlicht wurden und bis Dato nur mit der 32 KRAM liefen. Diese Erweiterung ist zwar überall käuflich zu erwerben, kostet aber zwischen 450 und 650 DM. Die hier vorgestellte Erweiterung beläuft sich auf nur 149 DM.
Selbst das Mini Mem, bisher der günstigste Einstieg in die Welt der Maschinensprache, kann da nicht mithalten. Falls Sie überhaupt noch irgendwo ein solches Modul erwerben können.
Diese kleine Hardwareschaltung läßt uns aufatmen. Der TI ist noch lange nicht tot. Jedenfalls solange nicht, wie solch gute Hardware auf den Markt dringt und auch in großen Stückzahlen verfügbar ist. Die Produktion läuft auf vollen Touren und wer sich solch ein Gerät zulegen will, der sei an die Firma RADIX Bürotechnik in Hamburg verwiesen. Sie werden mit Sicherheit nicht enttäuscht sein. Denn diese Hardware stellt eine echte Alternative zur Speichererweiterung oder dem fast nicht mehr zu bekommenden Mini Mem dar, hat fast den gleichen Befehlssatz, ist billiger und sehr einfach zu bedienen. *Klaus Weidemann*

BÖRSE

Hilfe! Die Austrittsgebühr beim Rex Soft Club beträgt 85 DM. Deswegen muß ich folgende Sachen verkaufen: Joysticks (35 DM), Joystickadapter (15 DM) + Othello-Modul (15 DM) + Statistik-Modul (20 DM). Tel. 07142/64321

Centronics-Interface (Extern) für TI 99/4A – 149,– DM. Tel. 07821/7570

---

6 aus 49 + 7 aus 38. 2 Systemprogr. (X-Basic) mit 28 Systemen. Kass. 30 DM + Nachnahme von P. Hielscher, Am Wall 22, 4401 Saerbeck

---

Suche: Floppy SS/SD – 150,–. o. DSDD 80 T. – 300 ggf. a. preisw. neu sowie Speechsynt. – 70,–. Tel. 07240/5300

Verkaufe TI Basic/Extended Basic Handbuch gegen Höchstgebot 4 Wo. n. Erscheinung. Vers. p. NN. Franz Tersluisen, Tel. 02851/7438 n. 17 Uhr

---

Salpha-Software und Module Donkey-Kont 60, Dig-Dug 45, Defender 35, Burgertime 50, Q-Bert 60, Re.to Pirate I.50, Parsec 20, Tombstone Ci. 20, Alpiner 30, Chisholm Tr. 20, Munch-Man 30, Ti-Invaders 20. Tel. 0208 23 299 tgl ab 18 Uhr

---

Kompl. TI-Station: Box, alle Karten, 2 Floppys, Drucker, Kass. Rec., Konsole mit integ. Grafic X-Basic, E/A-Modul, Adventure, Super Duper, Dsk. Manager, Speech-Synthesizer, 37er Color-TV, + Zub. 2999,– DM VB – Infos Tel.: 030/686 54 03

TI 99/4A in Orig.-verpackg. + X-Basic + 2 Joysticks + Rec-kabel + Literatur + PGMs (Lohn- und Einkommensteuer) 550,–. 040/2 70 30 78 Gert Adler (10.00 bis 15.00 Uhr)

Verkaufe TI99/4A + Exbasicmod. (original) + Schachmod. + Literatur. VB 500 DM (auch Einzelverkauf) Tel. 08039/26 30

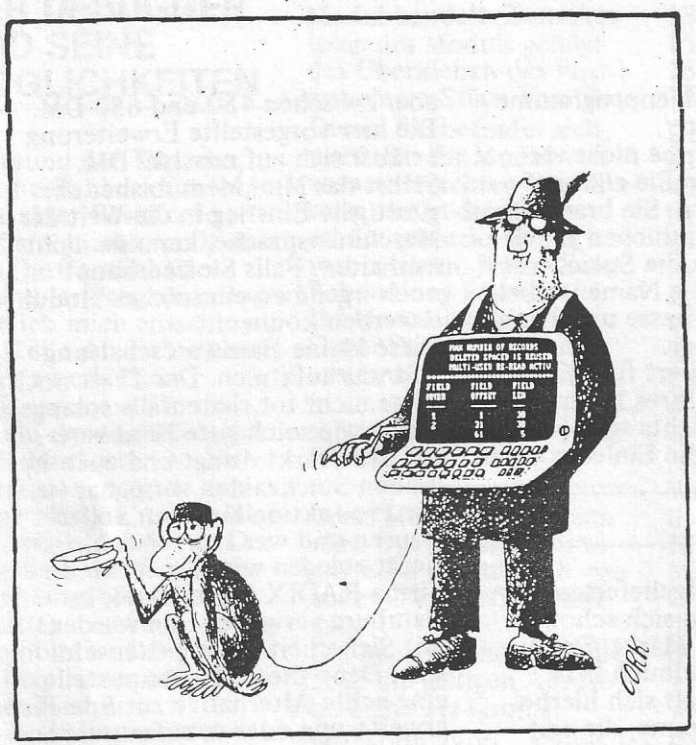
Konsole; Joysticks; Synthesizer; X-Basic; EA; MM; X-Box; Diskettenlaufw.; Akustikkop.; RS232; Module; haufenweise Spiele; Informationsmaterial; Adressen; Bücher; Schaltpläne; Listings; Wert weit über 5500 VB 4200; kaum benutzt 760 24 06 (München)

TI 99/4A, P-Box, RS232, Disk + Contr., 32k, P-Code + USCD-Pasc., RGB-Mod., Joyst, E/A + dt. Handb., PIO-Kabel + Datenverw.-Modul, 4 Spielmod., Recorderkabel, alles einwandfrei und original TI. + Recorder + 15 Disks + ...
Kompl. 3200,– DM VB. 040/601 48 83

Hallo TI-Freaks. Verkaufe TI 99 Netzteil (50,– DM). Recorderkabel (10,– DM) orig. TI 99/4A Schaltnetzteil zum Austausch (50,– DM), TI-Tastatur (50,– DM), Jockstik (20,– DM) Kassettenrecorder (40,– DM), TI-Modulator (80,–) TI-99/4A Netzteil (30,– DM)
Suche TI-Module. F. Haage, Rudolfstr. 9, 746 BL-Frommern 07433/35189

++ GOLF ++ (TI-99/4A + ExBasic) Meistern Sie die 18 Bilder!! Viel Geschicklichkeit und Augenmaß sind gefordert! TOP GARBGRAPHIK! Bis 10 Spieler! DISK nur 10,– DM Vorauskasse! B&A AHRENS, Am Hinkelstein 21, 6140 Bernsheim 1

Verk. TI99/4A + Box + Floppy + 27 Disks + Ext. Bas + 6 Mod. Parsec-Minus-Schach-Blacky. Adventure-Disk Manager + Kassetten + Ori.Lit + Data Becker + Hefte + Zubehör. VB 1100. H.v. Smolinski, Schillerstr. 96, 2080 Pinneberg 04101–27233 FarbMoni +450 DM



SPRITE-KONSTRUKTEUR

Fortsetzung von Seite 139

unbedingt vorher mittels SAVE (das ist ein Programm des Arbeitsmenues) „aufheben".

Set all:

SET ALL setzt alle Punkte. Abgesehen davon, daß das Zeichenfeld nach der Anwendung von SET ALL nicht weiß, sondern schwarz ist, hat dieses Programm die gleiche Eigenschaft wie CLEAR. Es ist sehr nützlich, wenn Sie Sprites mit großen Dunkelflächen definieren wollen.

Ready:

Bei der Anwendung von READY, dem letzten Punkt des Arbeitsmenues, kehren Sie in das Hauptmenue zurück. Das von Ihnen gezeichnete Sprite wird dabei nicht gelöscht. Sie können also gegebenenfalls wieder ins Arbeitsmenue mittels WORKING zurückkehren und Veränderungen an der Zeichnung vornehmen.

Das Hauptmenue

Wie bereits erwähnt, dient das Hauptmenue dazu, erstellte Sprites zu verwalten. Wenn Sie Ihren Joystick nach rechts drücken, gelangen Sie nicht in den Arbeitsmodus. Zeichnen können Sie nämlich nur, wenn Sie sich im Arbeitsmenue befinden.

Save

Mit SAVE können Sie fertige Sprites abspeichern und damit für spätere Verwendung aufheben.

Dabei fragt Sie der Computer nach dem Namen des Sprites. Dieser darf maximal 16 Buchstaben betragen. Drücken Sie die ALPHA-LOCK Taste und geben Sie den Namen ein. Nach Drücken von ENTER sagt Ihnen der Computer, das wievielte Sprite Sie soeben gesaved haben. Nicht der Spritename, sondern die Spritenummer ist ausschlaggebend für das „Zurückholen" mittels CREATE.

Wurden bereits 18 Sprites gesaved, so erscheint bei versuchtem Einstieg in das Programm SAVE die Fehlermeldung WARNING: MEMORY FULL. Achtung: Diese Fehlermeldung kommt in diesem Fall nicht vom Betriebssystem des TI-99/4A, sondern wird softwaremäßig simuliert. Tatsächlich haben Sie auch bei 18 gespeicherten Sprites noch etwa 500 Bytes freien Speicherraum zur Verfügung.

Create:

(Nur in der Version 2.6–85, sonst OLD ONE)

Mit CREATE können Sie alte Sprites, die Sie aufgehoben haben, wieder aufrufen. Der Computer fragt Sie nach der Spritenummer des Sprites. Geben Sie eine Nummer ein, die nicht existiert, so ignoriert der Computer die Eingabe. Bei der Eingabe einer gültigen Spritenummer stellt der Computer das aufgerufene Sprite im Zeichenfeld dar und schreibt den Spritenamen darunter. (Dieser dient in erster Linie zur Orientierung für Sie).

Wurde noch kein Sprite gesaved, ist der Einstieg in das CREATE-Programm nicht möglich!

Außerdem gibt es im Hauptmenue noch die Programme HEX OUT und LOOK. Sie sind im Hauptmenue nochmals angeführt, damit Sie bei Bedarf nicht dauern zwischen Haupt- und Arbeitsmenue wechseln müssen.

Die Funktion der beiden Programme ist im Hauptmenue genauso wie im Neben- oder Arbeitsmenue.

Erläuterungen, Programm technische Daten:

Das Unterbringen eines so umfangreichen Programmes in einen 16k-Speicher wurde erst durch die Verwendung zahlreicher Unterprogramme möglich. Dementsprechend oft tauchen daher im Programm die Wörtchen GOSUB oder CALL auf.

Ein weiterer Vorteil des Programmes besteht darin, daß auf komplizierte Mathematik verzichtet wurde. Bei jeder Funktion tastet der Computer den Bildschirm ab und holt seine Informationen unmittelbar vom Schirm, was sehr viel Speicherplatz spart.

Achtung: Da das Programm, wie bereits erwähnt, von der Mathematik weitgehend Abstand nimmt, bilden die Befehle CALL GCHAR und natürlich auch CALL HCHAR die Grundlage für die Funktion.

Bei der Eingabe des Programmes mittels Listing in den Computer sei daher darauf zu achten, stets die richtigen Variablen oder Zahlen in diesen beiden Befehlen einzusetzen. Abtippfehler können zu schweren Logikfehlern führen, welche auch sehr schwer zu eruieren sein dürften.

Nr. 1/85

Das erste Deutsche MSX-Magazin

MSX: DER STANDARD FÜR IHREN PRIVATEN PC

IM TEST:

Philips 8020
Spectravideo 728
Sony
Creative Graphics
Yashica 64
Philips Printer 0020
Ackobase
Ackotext
Sony Plotter C 41
Quickdisk QDM/01

LISTINGS:

32 Seiten
MSX-Programme

MARKTÜBER-SICHT:

Das komplette
MSX-Software-
Angebot!
Alle Fachbücher!

JETZT AN IHREM KIOSK

KAUFBERATUNG
SERVICE & SERIE

NEUE PRODUKTE FÜR TI-99/4A

Extended Basic mit deutschem Handbuch	DM 199,90
Extended Basic II Plus = Extended Basic + Grafic Extended Basic (Apesoft) in 1 Modul mit deutschem Handbuch	DM 299,–
Umbauaktion wir machen aus Ihrem Extended Basic (Mechatronic ein Extended Basic II Plus mit deutschem Handbuch für	DM 98,–
Umtauschaktion Bei Bestellung eines Extended Basic II Plus vergüten wir Ihnen bei kostenfreier Zusendung eines „Original amerikan. Extended Basic" Moduls (elektrisch/mechanisch einwandfreier Zustand!!!)	DM 70,–
32k-RAM-Erweiterung zum seitlichen Anstecken an den Bus, im Kunststoffgehäuse 190 x 110 x 60 mm, der Bus wird nach rechts durchgeschleift; mit 5V Steckernetzteil	DM 295,–
32k-RAM-Erweiterung, jedoch mit zusätzlichem Centronic-Interface nach hinten herausgeführt; Direktanschluß an PP-A4	DM 395,–
128k-RAM-Erweiterung (Ausführung wie 32k) mit Centronic-Interface	DM 595,–
Centronic-Interface im Gehäuse der 32k-RAM-Erweiterung	DM 188,–
4-Farben-Printer-Plotter PP-A4, Centronic-Schnittstelle, DIN-A4 Format, Direktanschluß an 32k-RAM	DM 699,–
Slim-Line-Laufwerk 5,25", 500kByte-DS/DD	DM 498,–
Einbausatz für 2 Laufwerke in Original-II-Box, mit Manual	DM 95,–
DISC-Steuerkarte (CorComp), DS/DD, bis zu 4 Laufwerke anschließbar	DM 635,–
Anschlußkabel PP-A4 an 32K-RAM	DM 68,–
QUICK-DISC-FLOPPY (im Gehäuse), zum Direktanschluß an TI-99/4A, mit 128kByte – DS, für 2,8" Disketten, mit Steckernetzteil	DM 598,–

Preise in DM/Stück inkl. Mwst.
Technische Änderungen vorbehalten
Versand gegen Nachnahme oder Vorauskasse.

albs-Alltronic G. Schmidt · Postfach 1130 · 7136 Ötisheim
Tel. 07041/2747 · Telex 7263738 albs

TI-99/4 A

32 K-Karte (Corcomp)	429,–
RS 232-Karte (Orig. TI)	429,–
P-Code-Karte (Orig. TI)	799,–
Compact Peripherie System 99 (Disccontroller DSD, 32 K-RAM, 2xV-23, 1 Centronics-schn.)	1200,–
dto. + 1 Laufw. DSDD	1698,–
Ext. 32 K-Erweiterung	299,–
dto. = 1 Centronics-schnittst.	359,–
Externes Druckerinterface + Kabel + Epson RX 80	1299,–
dto. + RX 80 FT+	1519,–
dto. + FX 80+	1819,–
dto. + Star SG-10	1279,–
2. Laufwerk extern Slimline DSDD mit Gehäuse + Netzteil	899,–
Sprachsynthesizer	189,–
Grafiktablett (Supersketch)	239,–
Modulexpander 3fach	125,–
8fach	225,–
Akustikkoppler Dataphon S 21 d + externe V-24-Schnitstelle + Verbindungskabel	579,–
Fernbedienung (Orig. TI)	65,–
Joystickinterface + 2 Joysticks	95,–
Quickshot II	29,–
Cassettenrecorderkabel	259,–
Extended Basic (Orig. TI)	239,–
Ext. Basic (dt. Nachbau)	299,–
Extended Basic II Plus	289,–
Mini Memory	175,–
Editor/Assembler (32 K notwendig)	299,–
TI-Writer (32 K notw.)	299,–
Multiplan (32 K notw.)	95,–
Terminal Emulator II	99,–
Speech Editor	je 29,–
Connect four, Yahtzee	je 39,–
Attack, Chisholm, Trail, Othello, Tombstone City, TI-Invaders	je 49,–
Parsec, Alpiner, Indoor, Soccer, Car Wars	je 59,–
Fathom, Microsurgeon, Moonmine, Hopper, Bigfoot, Statistik	je 69,–
Espial, Defender, Dig Dug, Tunnels of Doom, Star Trek	79,–
Datenverwalter + Analyse	
Moon Patrol, Jungle Hunt, Hs. Pacman, Donkey Kong, Buck Rogers, Burgertime, Congo Bongo, Return to Pirate's Isle, Treasure Isle, Adventuremodul	je 75,–
Video Chess, Moonsweeper	je 79,–
Popeye	je 89,–
Miner 2049	99,–

BÜCHER

Editor/Assembler Handbuch dt.	98,–
Extended Basic Handb. dt.	48,–
TI-Basic & Extend. Basic	48,–
Mini Memory Spezial dt.	55,–
Assemblerhandbuch für das Mini Memory dt.	78,–
TI-99/4A intern in dt. (Betriebssystem, GPL, ROM ...)	38,–

DISKETTEN- UND CASSETTENSOFTWARE

Superbasic, Exbasic II + Painter, Extended Basic Compiler, Graphicmaster, 3-D-World, Forth, Apesoft-Programme, TI-Text, TI-Datei, Skat ... a.A.

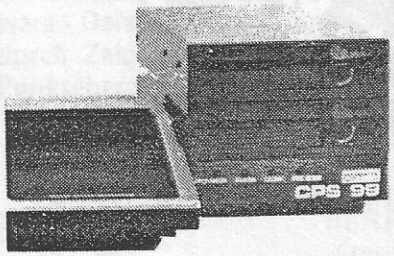
Alle Preise inkl. MWSt. zuz. Versandkostenpauschale (Warenwert bis DM 1.000,–/darüber): Vorauskasse (DM 8,–/20,–), Nachnahme (DM 11,20/23,20), Ausland (DM 18,–/30,–. Versand nur gegen Vorauskasse oder per NN; Ausland nur Vorauskasse. Gesamtpreisliste gegen Freiumschlag.

CSV RIEGERT

Schloßhofstr. 5, 7324 Rechberghausen,
Tel. (0 71 61) 5 28 89

Wir lassen den TI-USER nicht im Stich!

atronic



Die neue Ti-Peripherie von ATRONIC

● **CPS 99:** Das kompakte System!
Diskontroller (90—360 KByte je Laufwerk) 1—2 Laufwerke einsetzbar.
32 KByte RAM, 2 x RS 232—1 x parallel Schnittstelle

● **Expansion System**
Die kleinen Erweiterungen mit der großen Leistung
- 32 KByte RAM
- 32 KByte RAM + Centronics-Interface
- Centronics-Interface

Alle RAM-Erweiterungen in hochwertiger C-MO5-Technologie!

Weitere Information erhalten Sie bei jedem guten TI-Händler oder direkt von:

atronic
Meiendorfer Weg 7
2000 Hamburg 73
Tel. 0 40 / 6 78 93 08-09

special-ASSEMBLER-special

TI 99/4A SPECIAL

Nr. 3/86 DM 19,80 / ÖS 158 / SFR 19,80

Über 90 Seiten Alles über Assembler für den TI 99/4A

ASSEMBLER ASSEMBLER

MACHEN SIE MEHR AUS IHREM TI MIT ASSEMBLER

Aus dem Inhalt:

- DELETE
- BILDSCHIRM SPEICHERN
- SCROLL IN AUSSCHNITTEN
- HILFE FÜR DAS MINI MEM
- FASTCOPY
- BIT MAP MODE
- ASSEMBLER IN TI BASIC
- ZEICHENDEFINITION

ASSEMBLER

Ein Muß für jeden Assembler-Anwender!

ASSEMBLER

special-ASSEMBLER-special

TI REVUE

IMPRESSUM

TI ASSEMBLER ist eine Sonderpublikation der TI REVUE in der AKTUELL GRUPPE Werner E. Seibt.
Verantwortlich für den Inhalt: Heiner Martin
Verantwortlich für Anzeigen: Bruno G. Redase.
Alle: Postfach 1107 in 8044 Lohhof.
Anfragen bitte nur schriftlich.
Druck: Maier und Söhne
Es gilt die Honorarliste des Verlages.
Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung
Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für einen einmaligen Abdruck sowie die Aufnahme in den Programm-Service nach den Verlags-Sätzen!

Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Jedwede Verwertung ist untersagt, Nachdruck nur mit ausdrücklicher schriftlicher Zustimmung des Verlages. Namentlich gezeichnete Artikel geben nicht unbedingt die Meinung der Redaktion wieder.
Kein Anspruch auf Lieferung bei Ausfall durch höhere Gewalt.

Gerichtsstand: München
Geschäftsführer: Werner E. Seibt
Abo- und Kassetten-Service:
Henny Rose Seibt

© by TI/CBM Verlag
SPS und Autoren.

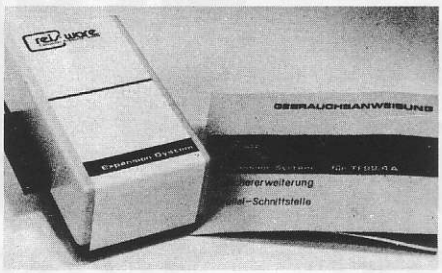
TI - 99/4A

Compact Peripherie System 99

CPS 99 mit einem Laufwerk 1.698,-
DS DD = 360 K mit 32 K-RAM,
2 x RS 232, Centronics Interface
Disk-Controller DS DD

CPS 99 mit zwei Laufwerken 2.198,-
DS DD = 720 K mit 32 K-RAM
2 x RS 232, Centronics Interface
Disk-Controller DS DD

Externe Erweiterungen



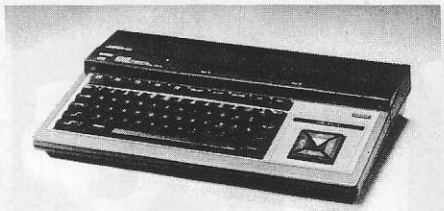
NEU - NEU - NEU - NEU - NEU - NEU

256 K Byte RAM-Expansion (RAM-Disk) 598,-
- **Ausbaubar bis 1 Megabyte**
- **Betrieb mit vorh. 32K Byte Erweiterung möglich**
- **Unterstützt Basic, Extended Basic u. Assembler**
- **Erweiterter Befehlsvorrat für Basic u. Ext. Basic**
- **Ultraschneller Zugriff auf bis zu 8 Programme durch RAM-Banking (bei 256K-Version)**
- **Wesentlich schnellere Bearbeitung von Disk-Files**
- **Schnittstelle für Softcard eingebaut**

Alle Preise incl. MwSt. zuzügl. 5,- DM Versandkosten. Lieferung per Nachnahme oder Vorkasse.
Ab 200,- DM versandkostenfrei.
Fordern Sie kostenlos unsere Sonderpreisliste an.

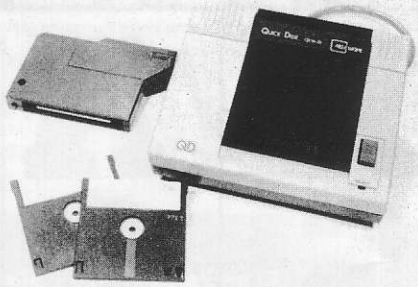
MSX MSX

MSX-Computer



Sanyo mpc 64 698,-
deutsche Tastatur, Resettaste und Einschalter obenliegend, 2 Modulslot

MSX-Zubehör



Disk-System 2,8" QDM-01 398,-
2,8" Diskette 2 x 64 K (Quick-Disk)
umfangreiche Software
1 Jahr Garantie

Disketten 2,8" (10 Stück) 89,-

Software auf Modul oder Quick-Disk ab 39,-

MSX-Einsteiger-System 898,-
bestehend aus:
1 MSX Computer Yashica YC-64
1 Disk-System 2,8" (Quick-Disk)

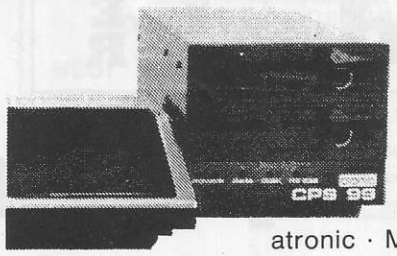
Für weiteres Zubehör und Software fordern Sie unsere kostenlose Preisliste.

Programm-Service REIS GmbH
D-5584 Bullay
Bergstraße 80
Telefon 06542/2715

Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr.
Für technische Fragen: 07 31/3 32 20 und
für Listings/Programme: 0 89/1 29 80 13

Wir lassen den TI-USER nicht im Stich!

atronic



- **CPS 99:** Das kompakte System! 2 × RS 232, 1 × Parallel Interface 32 KByte, Speichererweiterung Disketten-Controller/Disk-Drive
- **32 K RAM Erweiterung**
- **Centronics Interface**
- **V24 (RS 232) Interface**
- **32 K RAM + Centronics**
- **Externe Disk-Laufwerke**
- **RS 232/Centronics** Karte

KARTEN FÜR PERIPHERIE-BOX:

Neu: 256 K-Speichererweiterung (RAM-Disc)
- **Bis 1 MB ausbaubar, umfangreiche Software implementiert ✻ Für Basic + Ext + Assembler**
- **32 K RAM** Erweiterung
- **Disk-Controller** (bis zu 4 × 360 KByte)
- **Interface Karte mit 32 K RAM**
- **Controller Karte mit 32 K RAM**

FORDERN SIE DIE PREISLISTE AN!

atronic-Produkte bekommen Sie bei jedem guten TI-Händler oder direkt bei:
atronic · Meiendorfer Weg 7 · 2000 Hamburg 73 · Tel. 0 40 / 6 78 93 08-09 · Tx. 2 174 031

special - ASSEMBLER - special

TI 99/4A SPECIAL

SONDERHEFT

Nr. 3/86/März

INHALT

Grüß Gott - Gruezi - Guten Tag

Mit diesem Sonderheft halten Sie das erste der TI-REVUE in der Hand, welches sich ausschließlich mit Assemblerprogrammen beschäftigt. Ganz auf Basicprogramme konnten wir allerdings auch hier nicht verzichten, ein sehr komfortabler Basic-Assembler für das Mini-Memory ist in diesem Sonderheft auch enthalten. Für die Besitzer des Extended Basic und einer Speichererweiterung sind bei diesem Assembler alle notwendigen Änderungen angegeben, so daß sich das Programm auch damit einsetzen läßt. Es ist dann allerdings noch nicht durch Mehrfachanweisungen in jeder Zeile für das Extended Basic optimiert. Dies dürfte aber wohl einfach zu realisieren sein. Damit kann dann noch ein bißchen Geschwindigkeit herausgeholt werden.

Bei der Zusammenstellung der Programme haben wir uns bemüht, ein möglichst breites Spektrum verschiedener Anwendungen zu treffen. Auch an notwendige Utilities haben wir gedacht. Viele der Programme geben über den eigentlichen Anwendungszweck hinaus wertvolle Tips zur Assemblerprogrammierung. So sind zwei Utilities, die beim Extended Basic fehlen, DSRLNK und GPLLNK, in diesem Sonderheft in sehr kurzen Ausführungen enthalten. Anhand eines Sektorkopierprogramms wird der Zugriff auf einzelne Sektoren von Disketten gezeigt. Eine besondere Art der Speicherplatzreservierung im VDP-RAM unter Extended Basic ist im Programm interruptgesteuerte Soundlistenverarbeitung enthalten. Diese Reservierung arbeitet übrigens nach den bisherigen Erkenntnissen auch unter TI-Basic. Das waren nur einzelne Beispiele und wir glauben, daß Ihnen die Programme jede Menge Anregungen geben können.

Zum Schluß sei noch ein Wort zum Rücksprung aus dem Assembler-Programm in das Basic gesagt. Texas Instruments schreibt hier vor, daß das Condition-Bit im GPL-Statusregister gelöscht wird und dann eine Verzweigung zur Adresse >0070 geschieht. Nun sind viele Anwender darauf gekommen, daß dies einfacher mit einer direkten Verzweigung zur Adresse >006A geschehen kann. Dies muß, da diese Adresse nicht unbedingt garantiert ist, d.h. es kann einzelne TI 99/4A-Konsolen geben, bei denen dies nicht richtig arbeitet, eventuell jeweils geändert werden.

Wir hoffen, daß wir mit der Auswahl der Programme Ihren Geschmack getroffen haben und wünschen Ihnen eine weitere erfolgreiche Arbeit in Assembler.

Die Redaktion

SERVICE

Assembler — kein Problem
Leicht gemacht
Auf Seite 4
Im RAM liegt manches versteckt
Auf Seite 6
Ein bißchen Bit bitte
Auf Seite 8
Größer und kleiner —
jetzt wird es trickreich
Auf Seite 14
Der Schlüssel zum
Betriebs-System
Auf Seite 17
Service-Karten
Auf Seite 46
Börse
Ab Seite 89

LISTINGS

Pixelwolf
Auf Seite 20
Palindrom
Auf Seite 22
Interuptgesteuerte Sound-
listenverarbeitung
Auf Seite 24
Delete
Auf Seite 28
Bildschirm speichern
Auf Seite 30
Scroll
Auf Seite 34
Life
Auf Seite 35
Fastcopy
Auf Seite 36
GPLLNK
Auf Seite 40
Bit Map Mode
Auf Seite 41
Assembler in TI-Basic
Auf Seite 62
Hardcopy
Auf Seite 68
Char/Chad
Auf Seite 74

ASSEMBLER LEICHT GEMACHT FÜR DEN TI 99/4A

Auf den nächsten Seiten wollen wir ein bißchen näher auf das Thema Maschinensprache mit dem TI 99/4A eingehen. Dabei sollen nicht nur das Editor/Assembler Modul, sondern auch die Module Extended Basic und Mini-Memory angesprochen werden. Beginnen wollen wir ganz einfach, aber am Schluß jedes Beitrages sollen auch für die Könner einige spezielle Tricks verraten werden.

Was brauchen wir, um mit Assembler arbeiten zu können? Auf jeden Fall eines der eben schon angesprochenen Module, wobei hier das neue Extended Basic II plus zum Extended Basic zählt, auch wenn es eine besonders nützliche Routine zum Abspeichern von Assembler Programmen auf Kassette besitzt, ähnlich wie das Mini-Memory.

DAS MINI-MEMORY:

Mit diesem Modul kann man allein mit der Konsole zusammen schon in Maschinensprache (Assembler) programmieren. Eine eventuell vorhandene Speichererweiterung ist ebenfalls nutzbar. Ein Line-by-Line Assembler, also ein Programm, welches die Bildschirmeingaben in direkt vom 9900-Prozessor verwendbaren Code umwandelt, gehört zum Lieferumfang dieses Moduls. Zum Abspeichern der Assemblerroutinen genügt beim Mini-Memory ein Kassettenrekorder, ein Abspeichern auf Diskette ist nicht möglich, jedoch können mit dem Mini-Memory vom Editor/Assembler auf Diskette erstellte Maschinenprogramme verarbeitet werden.

DAS EXTENDED BASIC:

Bei diesem Modul ist unbedingt eine Speichererweiterung notwendig. Das Modul enthält auch kein Assembler-Programm, welches ebenfalls noch benötigt wird, allerdings wurden schon mehrere derartige Programme veröffentlicht. Zusammen mit einem Diskettenlaufwerk können vom Editor/Assembler erstellte Assemblerprogramme benutzt werden. Problematisch ist das Abspeichern der mit einem Extended Basic Assembler erstellten Routinen. Hier sind nochmals Hilfsprogramme notwendig.

DER EDITOR/ ASSEMBLER:

Er ist ohne Zweifel das komfortabelste Modul für die Assemblerprogrammierung, verfügt es doch über einen Editor, mit dem der Quellcode deutlich lesbar erstellt wird. Dieser Quellcode wird von einem Assembler, der auch einen kompletten Ausdruck auf einem Drucker erlaubt, in den Objektcode umgewandelt, welcher dann wieder zum Abarbeiten geladen wird. Allerdings erfordert dieses Modul auch die größte Ausrüstung: Speichererweiterung und Diskettenlaufwerk.

GRUNDSÄTZLICHES

Beim TI 99/4A wird Assembler nicht mit dem „Hammer programmiert", wie bei vielen anderen Computern dieser Klasse. Ursprünglich dachte Texas Instruments wohl nur an den Editor/Assembler, und hier ist das erstellte Maschinenprogramm so richtig komfortabel: Eine relative Adressierung ist möglich, d.h. der Benutzer muß sich nicht um den belegten Speicherplatz kümmern, sondern das besorgt das Programm, welches den Objektcode einfach in einen freien Speicherbereich lädt und alle notwendigen Adressen dabei umrechnet. Der Aufruf des Programms (LINK) erfolgt nun auch nicht über irgendeine Systemadresse, die man sich permanent merken muß, sondern über einen Namen, der bis zu 6 Stellen lang sein darf und der in einer gesonderten Tabelle zusammen mit der dazugehörigen Einsprungadresse abgelegt ist. Diese Benutzerfreundlichkeit bedarf aber nun einiger Übersicht bei direkter Assemblierung, wie es mit dem Line-by-Line Assembler des Mini-Memorys und mit fast allen Extended-Basic Assemblern geschieht.

DIE LITERATUR

Unabdingbar notwendig zum Programmieren des TI 99/4A in Assembler ist das Handbuch zum Editor/Assembler. Dies ist in Englisch geschrieben, es gibt davon aber auch eine zwar nicht als exzellent zu bezeichnende, aber doch hilfreiche Übersetzung ins Deutsche für diejenigen, die des Englischen nicht sehr mächtig sind. Eine weitere, sehr gute Lektüre zum Thema 9900 Assembler gibt es leider nicht mehr: Das 16 Bit Microprozessor Kursbuch von Texas Instruments. Ebenfalls sehr wichtig ist natürlich die Anleitung zum jeweils verwendeten Assembler (bei Extended Basic und Mini-Memory.

DAS ERSTE PROGRAMM

Aller Anfang ist schwer, besonders bei der Programmierung in Maschinensprache. Zwei der wesentlichen Schwierigkeiten mit Assembler wollen wir gleich zu Anfang behandeln: Der Einsprung auf das Programm und das Darstellen eines Zeichens auf dem Bildschirm. Folgendes Programm soll als Beispiel dienen:

```
        DEF   START
        REF   VSBW
MYWS    BSS   32
START   LWPI  MYWS
        LI    0,>0190
        LI    1,>4100
        BLWP  @VSBW
ENDLOS  LIMI  2
        LIMI  0
        JMP   ENDLOS
        END
```

Dieses Programm für das Editor/Assembler Modul wird unter dem Namen START aufgerufen, das beinhaltet die erste Zeile mit DEF. REF bedeutet Reference und ist eine weitere bedienungsfreundliche Funktion: Beim Laden des Programms wird

hier die REF/DEF-Tabelle abgesucht, ob der Begriff enthalten ist und wenn ja, wird der entsprechende Wert eingesetzt.
Dann werden mit BSS 32 Bytes (16 Worte) für den Workspace (Arbeitsbereich) des TMS 9900 reserviert. Dieser Prozessor besitzt ja nicht die sonst viel verbreiteten Register in der CPU, sondern benutzt dafür einen Bereich im Ram, auf den der Workspacecounter zeigt. Ab dieser Adresse werden 16 Register (R0 bis R15) benutzt. Als 16-Bit Prozessor umfassen diese Register natürlich je 2 Bytes, deshalb der Wert 32 bzw. >0020.
Das Programm beginnt damit, daß der Workspacecounter geladen wird, und zwar mit der Anfangsadresse des eben reservierten Speicherbereiches. Nun laden wir die Register R0 mit der Adresse, in die wir in das VDP-Ram schreiben wollen und das höherwertige Byte von R1 mit dem Wert >41, was dem ASCII Wert des Buchstabens A entspricht. Mit BLWP wird nun ein Unterprogramm aufgerufen. Solche Unterprogramme stellen alle Module, mit denen in Assembler gearbeitet werden kann, zur Verfügung, VSBW ist dabei eine Kurzform für VDP Single Byte Write, was soviel bedeutet wie ein Byte in das VDP-Ram schreiben.
Danach wird mit LIMI 2 der Interrupt freigegeben und mit LIMI 0 wieder unterbunden. Dies dient dazu, daß wir mit der Quit-Taste, die ja von der Interrupt-Routine abgefragt wird, aus der Endlosschleife zum Titelbild kommen können. Die Endlosschleife ist mit dem unbedingten Sprungbefehl JMP (Jump) realisiert.
Kommen wir nun zu den verschiedenen Assembler. Mit dem Editor/Assembler erstellen wir mittels des Editors die Quellcode-Datei und assemblieren diese dann mit dem Assembler. Dabei sollten wir die Dateien schon vom Namen her deutlich kennzeichnen, z.B. mit DSK1TESTQ (für Quellcode) und DSK1TESTC (für Objektcode). Bei der Abfrage OPTIONS des Assemblers geben wir RC oder nur R ein. Wird das C gewählt, wird ein sogenannter compressed Code erstellt, dieser kann vom Extended-Basic-Modul nicht geladen werden. Danach wählen wir den Abschnitt LOAD AND RUN, tippen DSK1.TESTC ein und starten das Programm über den Namen START.
Nun haben wir aber einen einfachen Assembler in Extended-Basic geschrieben oder den des Mini-Memorys. Hier sind ein paar Einschränkungen zu machen: DEF, REF und BSS sind meist nicht vorhanden. Wir beginnen also an einer freien Speicherstelle, merken uns die Adresse (z.B. >7000 für das Mini-Memory Adresse (z.B. >2500 unter X-Basic oder >7000 für das Mini-Memory) und springen 32 Bytes höher, d.h. auf >2500 bzw. >7D20. nun beginnen wir mit dem Programm:

```
LWPI    >2500 (>7D00)
LI      R0, >0190
LI      R1, >4100
BLWP    §>2020 (>6024)
LIMI    2
LIMI    0
JMP     >2530 (>7D30)
END
```

Die Werte in Klammern gelten für das Mini-Memory. Wir haben also für den Aufruf des Unterprogramms eine feste Adresse eingegeben und dies ebenfalls beim Wert für den Sprung getan.
Hier mußten wir uns während des Programmierens merken, auf welche Adresse wir springen wollten. Weiter haben wir uns die Adresse des Programmbeginns (>2520 bzw. >7D20) und der ersten freien Adresse hinter dem Programm gemerkt (>253A bzw. >7D3A);
Jetzt können wir aber das Programm noch nicht aufrufen und so müssen wir die dafür notwendigen Zeiger noch entsprechend laden.

Zuerst müssen wir das Programm gegen Überschreiben schützen und den Zeiger auf den Beginn des freien Rams laden. Beim Extended Basic ist die Adresse >2002 mit >253A und beim Mini-Mem ist die Speicherzelle >701C mit >7D3A zu laden. Am einfachsten geschieht dies auch mit dem Line-by-Line Assembler mit AORG>2002 (> 701C für Mini-Mem) und dann DATA >2536 (>7D36). Nun müssen wir noch die DEFinitionstabelle und den Zeiger darauf laden. Unter Extended-Basic ist dies die Speicherstelle >2004 (>701E für Mini-Mem). Von dem Wert, der bisher in dieser Speicherstelle steht, ziehen wir 8 ab und geben diesen Wert dann ein, also z.B.> 3FF8 bei Extended Basic oder >7FF8 bei Mini-Mem. Die letzte Stelle dieses Zeigers ist immer eine 8 oder eine 0. Dann gehen wir an die Adresse, deren Wert wir gerade eingegeben haben mit z.B. AORG> 3FF8 (>7FF8) und nun wird es kompliziert. Jetzt müssen wir mittels DATA den Namen in ASCII eingeben. Bei dem Namen Start ergibt sich >5354, >4152, >5420. Bitte beachten, der Name ist immer 6 Stellen lang, notfalls mit >20 (entspricht dem Leerzeichen) füllen. Darauf folgt nun die Startadresse des Programms also >2520 bzw.> 7D20 beim Mini-Mem.
Jetzt können wir das Programm mit dem Namen aufrufen und ausführen.

In Extended Basic bzw. im Basic geschieht dies mittels CALL LINK ("START"). Jetzt funktioniert das Programm über LOAD AND RUN des Mini-Mems und des Assembler-Moduls einwandfrei, wir sehen das A deutlich auf dem Bildschirm, aber aus dem Basic oder Extended Basic aufgerufen, sehen wir gar nichts. Das liegt an einer Eigenheit, wie im Basic der Bildschirm aufgebaut ist: Hier sind alle Zeichen mit einem Offset von> 60 abgebildet. Soll das Programm also unter Basic arbeiten, muß die entsprechende Zeile heißen:

```
LI    R1,>A100
```

Übrigens könnt Ihr mir auch gerne schreiben, wenn Ihr ganz besondere Probleme mit der Assembler-Programmierung habt, aber bitte nicht zur Fehlersuche in 20KByte langen Programmen! Auf die Anwendung des Debuggers werden wir später eingehen.

FÜR EXPERTEN

Bleibt noch unser Tip für die Assembler-Könner unter Euch: Im Betriebssystem der RS232-Karte steckt eine Interrupt-Routine, die ein interruptgesteuertes Empfangen von Zeichen erlaubt. Dazu müssen folgende Pointer gesetzt werden: Auf> 8300 befindet sich der Pointer zum PAB-Buffer im VDP, auf> 8302 (Byte) die maximale Datensatzlänge, auf> 8303 (Byte) die Länge des zu empfangenden Datensatzes und auf > 8304 (Byte) die aktuelle Länge des Datensatzes. Im Falle eines Empfangserrors ist das letzte empfangene Zeichen> FF. Hat der Datensatz die auf >8303 angegebene Länge erreicht, wird als empfangenes Zeichen> FE in den PAB Buffer geschrieben. Um das Ganze zu aktivieren, wird ganz normal eine Datei eröffnet, nur mit dem Unterschied, daß in der PAB beim OP-Code das 1. Bit (MSB) gesetzt werden muß, also z.B. 80 für OPEN. Benötigt wird das zum Beispiel für Terminal Emulator Programme.

IM RAM LIEGT MANCHES VERSTECKT

Der TMS 9900 besitzt keine internen Register, wie viele anderen Microprozessoren, sondern verwendet dazu 16 Register a 2 Bytes, die irgendwo im RAM liegen können. Auf den Anfang dieses Registersatzes zeigt der Workspacecounter (Arbeitsbereichszeiger). Dieser Workspacecounter kann mit dem Befehl LWPI geladen werden, wie wir es ja schon beim letzten Mal benutzten. Ein LWPI >2500 belegt also den Speicher wie folgt:

> 2500 Register 0
> 2502 Register 1
> 2504 Register 2
> 2506 Register 3
> 2508 Register 4
> 250A Register 5
> 250C Register 6
> 250E Register 7
> 2510 Register 8
> 2512 Register 9
> 2514 Register 10
> 2516 Register 11
> 2518 Register 12
> 251A Register 13
> 251C Register 14
> 251E Register 15

Prinzipiell können diese Register im Programm beliebig benutzt werden. Da einige aber auch besondere Bedeutung haben, sollten wir vorher schon einige Vereinbarungen treffen. So wird in R11 bei einem Sprung zu einem Unterprogramm mittels BL (Branch and Link, Sprung mit Sicherung der Rückkehradresse des Programmzählers) genau diese Rückkehradresse abgelegt, d.h. wir sollten R11 nicht anderweitig benutzen. R12 ist nur für CRU-Operationen (dazu kommen wir ein andermal) wichtig. Hier muß dann die CRU-Basisadresse enthalten sein. R13, R14 und R15 werden durch den Aufruf eines Unterprogramms mittels BLWP belegt, sind also in unserem Hauptprogramm verfügbar. Weiter kommt noch R0 besondere Bedeutung bei Schiebeoperationen zu, das soll uns aber noch nicht interessieren.

UNTERPROGRAMME

Auch bzw. gerade in Assembler ist eine „Rattenschwanzprogrammierung" ein sehr schlechter und auch unübersichtlicher Programmierstil. Dennoch ist so etwas leider häufig zu sehen. Dabei erleichtern Unterprogramme, vor allem wenn man viel programmiert, die Arbeit.
Für eine bestimmte Lösung nimmt man nur das entsprechende Unterprogramm aus der Bibliothek. Dazu sollte man sich aber vorher die Benutzung der einzelnen Register genau überlegen.
Üblicherweise benutzen wir R0 bis R2, da diese auch viel von den Hilfsroutinen wie VMBW usw. benutzt werden, für Werte, die sich dauernd ändern. Damit wir für eigene Unterprogramme noch ein bißchen mehr zur Verfügung haben, nehmen wir dafür auch noch R3 und R4. R5 und R6 können für Schleifenzähler reserviert werden, R8 und R9 bleiben dann für Werte, die sehr häufig benötigt werden und R10 verwenden wir als weiteren Speicher für den Rücksprung. Dann können wir in zwei Unterprogrammebenen nach folgendem Muster arbeiten:

```
Hauptprogramm:
  ....      BL @XX
  ....
XX  MOV R11,R10
    .
    BL   @YY
    B    *R10
YY  .
    .
    RT
```

In dem Unterprogramm YY darf dann R10 natürlich nicht verändert werden. Die oben angeführte Aufteilung der Register ist natürlich nur als Vorschlag zu werten. Je nach den Notwendigkeiten des Programms wird sie entsprechend geändert.
Übrigens kann man bei einigen der Assemblerprogrammen bei den Registern das R auch weglassen, MOV R11,R10 kann man dann auch schreiben als MOV 11,10. Schreibfaule Leute machen das immer so. Es spart halt etwas Schreibarbeit.
Bei manchen Assemblerprogrammen können Labels, das sind solche Bezeichnungen wie XX und YY in dem Beispiel, nicht verwendet werden. Dann sind die Unterprogramme dem Hauptprogramm bei der Eingabe voranzustellen, damit wir uns jeweils die Startadresse des Unterprogrammes merken und dann im Hauptprogramm einsetzen können, z.B. als BL @>2500.

TEXTE AUF DEN BILDSCHIRM SCHREIBEN

Bei jedem Programm brauchen wir ja den Dialog mit dem Benutzer. Dazu müssen Informationen auf den Bildschirm geschrieben werden.
Hier sind dann in Assembler meist endlose Reihen von LI und BLWP zu sehen. Das Ganze läßt sich aber auch in einem Unterprogramm lösen:

```
TEXT1 DATA >0002
                (Adresse)
      BYTE 16 (Länge)
      TEXT 'Das ist ein
           Test'
TEXT  MOV   *R1+,R0
      MOVB  *R1+,2
      SRL   R2,8
      BLWP  @VMBW
      RT
```

Im Hauptprogramm muß dann das Unterprogramm wie folgt aufgerufen werden:

```
      .
      .
      LI   R1,TEXT1
      BL   @TEXT
      .
      .
```

Der Text ist wie folgt aufgebaut. Erst kommen zwei Bytes (ein Wort), welches die Bildschirmadresse enthält, wo der Text hingeschrieben werden soll. Dann kommt ein Byte, das die Länge des Textes enthält und dann der Text.
Achtung: Soll das Programm vom Basic heraus aufgerufen werden, so ist der Screenoffset von >60 zu berücksichtigen, d.h. wir müssen alle Bytes einzeln über DATA oder BYTE eingeben mit addiertem Offset. Ähnliches gilt bei Assemblerprogrammen, die die Anweisung TEXT nicht kennen, hier muß ebenfalls einzeln (mit oder ohne Offset) eingegeben werden.
Das Unterprogramm TEXT bereitet zuerst die Register für die Ausführung von VMBW vor.
R1 enthält die Startadresse des Textes im Ram.
So wird also zuerst R0 mit der Bildschirmadresse geladen. R1 wird dabei automatisch um 2 erhöht. Dann brauchen wir noch in R2 die Anzahl der zu übertragenden Bytes, also die Länge des Textes.
MOVB bewegt ein Byte in das höherwertige Byte von R2, SRL R2,8 ist ein Schiebebefehl und schiebt das Byte in das niederwertige Byte von R2. Mit der Hilfsroutine VMBW wird dann der Text in das VDP-

Ram geschrieben. Mit kleinen Änderungen kann dieses Unterprogramm auch in Schleifen verwendet werden, ohne daß im Hauptprogramm dauernd LI R1 steht.

SCHLEIFEN

Einige Leser fragten auch schon nach der Realisierung von FOR-NEXT-Schleifen in Assembler. Nun, das geht ganz einfach:

```
        LI   R5,200
LOOP    .
        .
        DEC R5
        JNE LOOP
```

Zuerst wird ein Register, hier R5, mit der Anzahl der Schleifendurchläufe geladen. Dann folgt das, was wir in der Schleife machen wollen. Abschließend wird mit dem Befehl DEC der Inhalt von R5 jeweils um 1 erniedrigt. Solange nicht 0 erreicht wird (beim Befehl DEC wird das Equal-Bit im Statusregister des Prozessor gesetzt, wenn das Ergebnis 0 ist), führt der Befehl JNE (Springe wenn nicht gleich) in die Schleife.

DER RÜCKSPRUNG ZU BASIC

Wollen wir am Ende eines Maschinenprogramms wieder zu dem Programm zurückkehren, aus dem das Maschinenprogramm aufgerufen wurde, so ist das ganz einfach zu realisieren:

```
CLR   @ >837C
LWPI    >83E0
      @ >0070
```

Dieser Rücksprung führt in den GPL-Interpreter zurück, wobei zuerst das GPL-Statusbyte gelöscht wurde, um anzuzeigen, daß kein Fehler aufgetreten ist. Dann wird wieder der GPL-Workspace geladen und das Programm, welches die Maschinenroutine aufgerufen hat, weiter ausgeführt.

FÜR EXPERTEN: AUTOSTART IN X-BASIC

Der Editor/Assembler und das Mini-Memory kennen einen Autostart des Maschinenprogramms direkt danach, wenn es von der Diskette geladen worden ist. Der Lader des Extended Basic kennt diesen Befehl leider nicht. Er läßt sich aber einfach nachbilden: Der Trick funktioniert über den User defined Interrupt. Die letzten Zeilen unseres Assembler-Programms (nicht mit einem Line-by-Line-Assembler erstellt) heißen dann:

```
AORG >83C4
DATA START
```

Damit wird als letztes im Ladevorgang die anwenderdefinierte Interrupt-Routine mit der Startadresse unseres Maschinenprogramms geladen. Wenn jetzt ein Interrupt ausgeführt wird, was ja alle 50stel Sekunde erfolgt, wird das Programm ausgeführt. In unserem Programm müssen wir dann nur ganz am Anfang noch einfügen:

```
CLR @>83C4
```

Damit nicht aus Versehen, wenn wir Interrupts zulassen, wieder von vorne begonnen wird.

EIN BISSCHEN BIT BITTE

Eine Anregung, etwas näher auf das Statusregister des TMS 9900 einzugehen, möchte ich aber gerne aufgreifen. Zuvor aber noch etwas anderes: Die Hilfsroutinen. Prinzipiell lassen sich beim TI 99/4A alle Maschinenprogramme auch unter allen Modulen, mit denen dies möglich ist, einsetzen. Eventuell müssen sie in einen anderen Speicherbereich gelegt werden, aber das ist nur bei einem Assembler, der mit absoluter Adresse arbeitet, wichtig. Ein Hindernis sind aber die von den einzelnen Modulen zur Verfügung gestellten Hilfsroutinen, die mittels des Befehls BLWP ausgeführt werden. Sie,bzw. die Vektoren darauf, stehen bei jedem Modul woanders. In der Tabelle 1 sind die Adressen zum Vergleich. Dabei ist nur zu berücksichtigen, daß das XMLLNK des Extended-Basic anders funktioniert als bei den beiden anderen Modulen.

Wollen wir also bei einem Programm ein Byte in das VDP-Ram schreiben, so ist bei einem Programm für das Mini-Memory zu schreiben:

```
LI    R0,>0100(VDP-Adresse)
LI    R1,>3100(BYTE)
BLWP@>6024
```

Für das Extended Basic müßte dies in einer Zeile geändert werden:

```
BLWP@>2020
```

Für das Editor/Assembler-Modul ist dann noch eine andere Adresse notwendig. Hier arbeiten wir aber besser mit der REFerence. Dieses REF bedeutet, daß beim Laden des Maschinenprogramms von Diskette eine REF-Tabelle nach dem Begriff, der in der Definition hinter REF genannt wird, abgesucht und der in dieser Tabelle enthaltene Wert dafür im Maschinenprogramm eingesetzt wird. Für das E/A-Modul schreiben wir also an den Programmanfang in den Quellcode:

```
REF   VSBW
```

und dann

```
BLWP@VSBW
```

Dies hat den Vorteil, daß dieses Programm dann auch ohne Änderungen für das Mini-Memory verwendet werden kann, da auch das Mini-Memory über die REFerence-Tabelle verfügt und beim Laden des Maschinenprogramms von Diskette dann eben die fürs Mini-Mem richtigen Werte eingesetzt werden. Leider geht das nicht fürs Extended Basic.

DAS EQUAL-BIT IM STATUSREGISTER

Kommen wir nun zum Statusregister des TMS 9900, wobei gleich vorab gesagt sei, daß sich dieses Thema nicht in einer Folge erläutern läßt, wir also immer wieder darauf zurückkommen werden. Der TMS 9900 besitzt neben dem Workspacecounter, den wir das letzte Mal besprochen haben, einen Programmcounter, das ist ein Register, welches die Adresse des nächsten abzuarbeitenden Befehls enthält, und ein Statusregister. Dieses wird durch das Ergebnis von vielen der Befehle beeinflußt, es werden je nach Ergebnis einzelne Bits gesetzt. Heute soll uns nur einmal das Bit 2 interessieren, auch als Equal-Bit bezeichnet. Der Name sagt es schon, dieses Bit wird gesetzt, wenn irgend etwas „Gleich" ist. Dies gilt für alle Vergleiche, also C (Compare = Vergleiche), CB (Vergleiche Byte) und COC, CZC und TB (diese sind etwas komplizierter, zu diesen kommen wir aber später noch). Bei allen anderen Operationen wird das Ergebnis 0 verglichen und wenn es 0 ist, dieses Bit ebenfalls gesetzt.
Betrachten wir also nochmal die Schleife:

```
   LI   R5,200
LOOP .
        DEC R5
   JNE LOOP
```

Der Befehl DECrement erniedrigt jeweils den Inhalt von R5 um 1. Erreicht dieser 0, wird das Statusbit für Equal gesetzt und der nachfolgende Sprung-Befehl JNE (Jump not equal) unbeachtet gelassen. Daraus sehen wir deutlich, daß die folgendes

```
DEC   R5
CI    R5,>0000
JNE   LOOP
```

zwar das gleiche Ergebnis bringt, aber eben einen überflüssigen Befehl enthält. Hier wird durch den (eben überflüssigen) Befehl CI (Vergleiche unmittelbar) der Inhalt von R5 mit 0 verglichen.

AUSGABE EINER HEX-ZAHL AUF DEN BILDSCHIRM

Ein weiteres Beispiel soll die Darstellung einer hexadezimalen Zahl in Dezimal auf dem Bildschirm sein. In Maschinenprogrammen wird ja häufig nur mit Ganzzahlen gerechnet, meist mit einem Registerinhalt. Mittels folgendem Unterprogramm kann dies realisiert werden:

```
   LI   R0,>0100 (Adresse)
   LI   R3,>0300 (Zahl)
   BL @HEXD
D10 DATA >000A

HEXD CLR   R2
     DIV  @D10,R2
```

```
MOV   R3,R1
SWPB  R1
AI    R1,>3000
BLWP  @ VSBW
DEC   R0
MOV   R2,R3
JNE   HEXD
B
```

Wir finden darin einen neuen Befehl, DIV (divide = teilen). Bei diesem ist zu beachten, daß für den Senkenoperanden 2 Register zur Verfügung stehen müssen, deshalb wird zuerst R2 auf 0 gesetzt mit CLR (Clear = Löschen). Dann erfolgt die Division durch 10. Als Ergebnis steht nun in R2 der Quotient und in R3 der Rest. Diesen Rest müssen wir nun als Zahl auf den Bildschirm darstellen. Um die Routine VSBW benutzen zu können, muß in R0 die Bildschirmadresse, das wurde vom Hauptprogramm erledigt, und im höherwertigen Byte von R1 das zu schreibende Byte stehen. Dies wird durch MOV und DWPB (Swap Byte = tausche Byte) erledigt. Nun müssen wir mit AI (Unmittelbare Addition) noch einen lesbaren ASCII-Wert daraus machen. Für Programme, die aus dem Basic gestartet werden, muß hier >9000 stehen. Danach folgt die neue Bildschirmadresse (nächste Stelle der Zahl), und dann werden mit MOV wieder die Register für die Division vorbereitet. Der Trick dabei ist, daß auch bei MOV das Equal-Bit des Statusregisters gesetzt wird, wenn das Ergebnis, d.h. hier der Inhalt von R3, Null ist. Gibt es also nichts mehr, durch das geteilt werden könnte, dann wird der nachfolgende Sprung nicht mehr ausgeführt und es erfolgt der Rücksprung mit B.

Für Experten:

Kassettenzugriff aus Assembler

In einer TI-Ausgabe hatten wir bei der Beantwortung von Leserfragen angegeben, daß es ein Leichtes sei, aus Assembler heraus die Adventure-Spiele auf Diskette zu überspielen. Wie mehrere Anrufe und Briefe deuten, gibt es da aber auch bei erfahrenen Programmierern Probleme. Es scheint auch so, daß hier einmal in einem Buch eine falsche Routine veröffentlicht wurde. Also, die Kassettenroutine ist eigentlich im Assembler-Handbuch und im Minimem-Handbuch ausreichend beschrieben. Wichtig ist nur noch eines: Auf FAC (>834A) muß der Name in ASCII stehen, also z.B. CS1. Das kann wie folgt realisiert werden:

```
NAME   TEXT'CS1'
       MOV  @NAME,@FAC
       MOVB @NAME+2@FAC
```

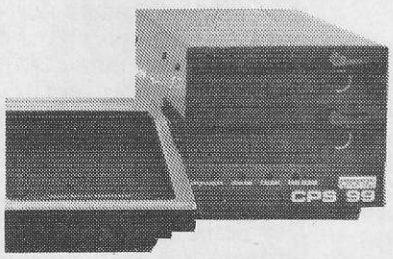
Ansonsten braucht man sich nur an die Anweisungen in den genannten Handbüchern zu halten.

Tabelle 1:
Equates für die Hilfsroutinen:

Bez.	E/A	Mini.mem	Ex-Basic	Beschr.
VSBW	>210C	>6024	>2020	Schreibt ein Byte in VDP-Ram
VMBW	>2110	>6028	>2024	Schreibt mehrere Bytes in VDP-Ram
VSBR	>2114	>602C	>2028	Liest ein Byte aus VDP-Ram
VMBR	>2118	>6030	>202C	Liest mehrere Bytes aus VDP-Ram
VWTR	>211C	>6034	>2030	Schreibt in VDP-Register
KSCAN	>2108	>6020	>201C	Tastaturabfrage
XMLLNK	>2104	>601C	>2018	Ausführen einer Betriebssystem-Routine
DSRLNK	>2120	>6038	n.a.	Aufrufen einer DSR-Routine
GPLLNK	>2100	>6018	n.a.	Ausführen einer GPL-Routine
LOADER	>2124	>603C	n.a.	Laden von Assembler-Programmen
NUMASG	n.a.	>6040	>2008	Wert an num. Basicvariable
NUMREF	n.a.	>6044	>200C	Wert von num. Basicvariable holen
STRASG	n.a.	>6048	>2010	Wert an String-Basicvariable
STRREF	n.a.	>604C	>2014	Wert von String-Basicvariable
ERR	n.a.	>6050	>2034	Rueckkehr Basic mit Error

Wir lassen den TI-USER nicht im Stich!

atronic



Die neue Ti-Peripherie von ATRONIC

● **CPS 99:** Das kompakte System! Diskontroller (90—360 KByte je Laufwerk) 1—2 Laufwerke einsetzbar. 32 KByte RAM, 2 x RS 232—1 x parallel Schnittstelle

● **Expansion System**
Die kleinen Erweiterungen mit der großen Leistung
* 32 KByte RAM
* 32 KByte RAM + Centronics-Interface
* Centronics-Interface

Alle RAM-Erweiterungen in hochwertiger C-MOS-Technologie!

Weitere Inforamtion erhalten Sie:
bei jedem guten TI-Händler, oder direkt von:

atronic
Meiendorfer Weg 7
2000 Hamburg 73
Tel. 0 40 / 6 78 93 08-09

Assembler leicht gemacht

Zur Erinnerung wollen wir hier das Beispielprogramm nochmals wiederholen. Es diente zur Ausgabe einer Hex-Zahl bzw. Registerinhaltes als Dezimalzahl auf dem Bildschirm.
Wie schon erklärt, finden wir darin einen neuen Befehl:

```
LISTING 1:

        LI    R0,>0100
        LI    R3,>0318
        BL    @HEXD



D10     DATA  >000A

HEXD    CLR   R2
        DIV   @D10,R2
        MOV   R3,R1
        SWPB  R1
        AI    R1,>3000
        BLWP  @VSBW
        DEC   R0
        MOV   R2,R3
        JNE   HEXD
        B     *11
```

DIV (divide=teilen). Bei diesem ist zu beachten, daß für den Senkenoperanden 2 Register zur Verfügung stehen müssen, deshalb wird zuerst R2 auf 0 gesetzt, mit CLR (Clear=Löschen) ; in R3 ist ja die Hexzahl, die ausgegeben werden soll, enthalten. Dann erfolgt die Division durch 10. Diesen Wert stellen wir als getrenntes DATA zur Verfügung. Als Ergebnis steht nun in R2 der ganzzahlige Quotient und in R3 der Rest. Diesen Rest müssen wir nun als Zahl auf den Bildschirm darstellen.
Um die Routine VSBW benutzen zu können, muß in R0 die Bildschirmadresse stehen, das wurde vom Hauptprogramm erledigt, und im höherwertigen Byte von R1 das zu schreibende Byte. Dies wird durch MOV und SWPB (Swap Byte=tausche Byte) erledigt. Nun müssen wir mit AI (Unmittelbare Addition) noch einen lesbaren ASCII-Wert daraus machen. Für Programme, die aus dem Basic gestartet werden, muß hier >9000 stehen, um den Screenoffset auszugleichen. Danach folgt die neue Bildschirmadresse (nächste Stelle der Zahl), und dann werden mit MOV wieder die Register für die Division vorbereitet. Der Trick dabei ist, daß auch bei MOV das Equal-Bit des Statusregisters gesetzt wird, wenn das Ergebnis, d.h. hier der Inhalt von R3 Null ist. Gibt es also nichts mehr, durch das geteilt werden könnte, dann wird der nachfolgende Sprung nicht mehr ausgeführt und es erfolgt der Rücksprung mit B.
Ein weiterer Punkt, der häufig zu Unsicherheiten führt, ist die Darstellung negativer Zahlen, wie sie der TMS 9900 verwendet. Manch einer wird sich auch schon gefragt haben, warum bei CALL PEEK und CALL LOAD manchmal eine negative Zahl als Adressenangabe zu verwenden ist. Dies hat seine Begründung in der Darstellung von negativen Zahlen für die CPU.
Hier müssen wir eine kleine Ausführung über das Zahlensystem unseres Computers einfügen. Bekanntlich kann der Rechner nur Einsen und Nullen (binäres System) unterscheiden, während wir im sogenannten Dezimalsystem rechnen. Nun lassen sich diese Zahlen einfach umrechnen. Darüber gibt die Tabelle 1 Aufschluß. Darin sind ebenfalls die Hexadezimalen Zahlen aufgeführt. Diese sind quasi aus der Notwendigkeit entstanden, nicht mit ewig langen Kolonnen von binären Zahlen arbeiten zu müssen. Ein Byte enthält bekanntlich 8 Bit und statt 00011010 schreibt sich wohl besser >1A, dabei dient das Größer-Zeichen zur Erkennung als Hexadezimalzahl. Als 16 Bit-Prozessor arbeitet der TMS 9900 sogar mit 16 Bit (2 Byte) langen Worten, das ergibt dann eine vierstellige Hexadezimale Zahl.

Tabelle 1:

Binär	Hex	Dezimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Kommen wir aber zu dem Vorzeichen der Zahl zurück. In nahezu allen Befehlen arbeitet die CPU mit folgender Zahlendarstellung: Das 1. Bit (MSB =höchstwertigstes Bit) der insgesamt 16 Bit langen Zahl bestimmt das Vorzeichen. Ist es 0, so handelt es sich um eine positive Zahl. Ist es 1, dann ist sie negativ. Dabei ist die negative Zahl das Zweier-Komplement der positiven Zahl. Einige Beispiele:
>0000 = 0
>0001 = 1
>7FFF= 31767
>8000 = 32768 (erstes Bit gesetzt!)
>FFFE= −2
>FFFF= −1

Wozu dient nun das Alles? Nun, das Statusregister des TMS 9900 hatten wir schon erwähnt und uns bisher ausschließlich mit dem Equal-Bit in diesem Register beschäftigt. Dieses Equal-Bit ist immer dann auf 1 gesetzt, wenn das Ergebnis einer Operation 0 war oder wenn bei einem Vergleich die Operanden gleich waren. In dem Statusregister beinhalten aber nach den meisten Operationen – (es gibt auch wenige, die das Statusregister nicht beeinflussen, darüber gibt aber das Handbuch zum Editor/Assembler Modul Auskunft) – noch andere Bits wichtige Informationen. Zwei davon wollen wir uns heute noch etwas näher ansehen: Das Logical Greater Bit (Logisch größer) und das Arithmetic Greater Bit (arithmetisch größer). Bleiben wir zuerst beim Letzteren, dem Arithmetisch Größer Bit. Wie der Name schon sagt, wird hier ein Vergleich nach den arithmetischen Regeln ausgeführt, d.h. das Vorzeichen der Zahl wird entsprechend berücksichtigt. Mit der Vergleichsoperation CI (Vergleiche unmittelbar) läßt sich die Auswirkung am einfachsten zeigen:

Der Befehl LI beeinflußt übrigens das Statusregister (im Gegensatz zu CLR), sodaß bei den letzten

```
LISTING 2:
        LI    R1,>1000   *DEZ 4096
        CI    R1,>0500   *DEZ 1280

ARITHMETRIC GREATER BIT gesetzt

        LI    R1,>0480   *DEZ 1152
        CI    R1,>0500   *DEZ 1280

ARITHMETRIC GREATER BIT nicht gesetzt

        LI    R1,>FFFE   *DEZ -1
        CI    R1,>F000   *DEZ -4096

ARITHMETRIC GREATER BIT gesetzt

        LI    R1,>E000   *DEZ -8192
        CI    R1,>F000   *DEZ -4096

ARTHMETRIC GREATER BIT nicht gesetzt

        LI    R1,>FFFE   *DEZ -1
        CI    R1,>0500   *DEZ 1280

ARITHMETRIC GREATER BIT nicht gesetzt

        LI    R1,>1000   *DEZ 4096
        CI    R1,>F000   *DEZ -4096

ARTHMETRIC GREATER BIT gesetzt

        LI    R1,>E000   *DEZ -8192
        MOV   R1,R1

ARTHMETRIC GREATER BIT nicht gesetzt

        CLR   R1         *DEZ 0
        MOV   R1,R1

ARITHMETRIC GREATER BIT nicht gesetzt

        LI    R1,>1000   *DEZ 4096
        MOV   R1,R1

ARTHMETRIC GREATER BIT gesetzt
```

Beispielen die MOV-Operation entfallen kann. Bei dieser wird ja der Registerinhalt des Senkenoperanten mit 0 verglichen und so soll hier nur gezeigt werden, wie auf einfache Art immer die entsprechenden Statusbits gesetzt werden können.
Für dieses Arithmetric Greater Bit stehen uns nun zwei bedingte Sprünge zur Verfügung. Bei JGT (Jump greater than) wird der Sprung ausgeführt, wenn das Bit gesetzt ist und bei JLT (Jump less than) wird der Sprung ausgeführt, wenn das Bit nicht gesetzt ist und das Equal-Bit ebenfalls 0 ist. Kommen wir also zu dem Logical Greater Bit. Dieses Bit wird im Statusregister immer dann gesetzt, wenn etwas logisch größer ist, d.h. ohne Beachtung eines Vorzeichens, das 16. Bit gehört dabei zur Zahl.

Auch hier wieder einige Beispiele zum besseren Verständnis:

```
LISTING 3:
        LI    R1,>1000   *DEZ 4096
        CI    R1,>0500   *DEZ 1280

LOGICAL GREATER BIT gesetzt

        LI    R1,>0480   *DEZ 1152
        CI    R1,>0500   *DEZ 1280

LOGICAL GREATER BIT nicht gesetzt

        LI    R1,>FFFE   *DEZ -1
        CI    R1,>F000   *DEZ -4096

LOGICAL GREATER BIT gesetzt

        LI    R1,>E000   *DEZ -8192
        CI    R1,>F000   *DEZ -4096

LOGICAL GREATER BIT nicht gesetzt

        LI    R1,>FFFE   *DEZ -1
        CI    R1,>0500   *DEZ 1280

LOGICAL GREATER BIT gesetzt

        LI    R1,>1000   *DEZ 4096
        CI    R1,>F000   *DEZ -4096

LOGICAL GREATER BIT nicht gesetzt

        LI    R1,>E000   *DEZ -8192
        MOV   R1,R1

LOGICAL GREATER BIT gesetzt

        CLR   R1         *DEZ 0
        MOV   R1,R1

LOGICAL GREATER BIT nicht gesetzt

        LI    R1,>1000   *DEZ 4096
        MOV   R1,R1

LOGICAL GREATER BIT gesetzt

      -      )UUUUUUUUUU
```

Selbstverständlich gibt es auch einige Sprungbefehle, die vom Logical greater Bit beeinflußt werden.
Bei JH (Jump high) wird der Sprung ausgeführt, wenn dieses Bit gesetzt ist, bei JL (Jump Low) wenn es nicht gesetzt wird. JHE (Jump high or equal) und JLE (Jump low or equal) berücksichtigt dann jeweils auch das Equal-Bit.

ASSEMBLER-KURS

Auf der vorigen Seite mußte der kleine Tip für die Experten entfallen. **Das waren dann die berühmten 10 Zeilen zuviel beim Umbruch, d.h. 10 Zeilen wären über die Seite hinausgegangen und „Gummi-Seiten" gibt es leider noch nicht.**
Deshalb diesen Tip nun hier am Anfang: Bei der Dateibehandlung unterscheidet der TI 99/4A ja 5 verschiedene Formate: Program, DIS/VAR, DIS/FIX, INT/VAR und INT/FIX. Gemäß dem Handbuch zum Editor-Assembler kann über die Abfrage des Status einer Datei auch das Format in Erfahrung gebracht werden. Dazu wird einfach ganz normal die Datei über DSRLNK angesprochen. Der Op-Code im PAB ist dafür dann >09. Hier ist in den meisten Disk-Controlern ein Softwarefehler enthalten. DIS/FIX-Dateien werden bei der Statusabfrage als DIS/VAR zurückgegeben.
Nun wollen wir aber weiter in unserem Assemblerkurs fortfahren. Viele Anfragen erhielt ich nach der Belegung des VDP-Ram's, d.h. was steht dort wo. Dies soll dann auch heute unser Thema sein. Vorab muß aber dazu gesagt werden, daß sich die Belegung je nach Modul unterscheidet. Wir wollen uns heute auf TI-Basic, Extended Basic und das Editor/Assembler-Modul unter der Funktion LOAD AND RUN beschränken, dabei ist das Mini-Memory unter LOAD AND RUN gleich wie das Assembler-Modul.
Rein für die Bildschirmdarstellung braucht der Video-Display Prozessor einige Tafeln. Im einzelnen sind dies:
1. Screen Image Table, auch als Bildschirmtabelle zu bezeichnen. In diesem Bereich stehen die abzubildenden Zeichen in ASCII. Dabei ist die linke obere Ecke des Bildschirms der Anfang der Tafel, der Buchstabe rechts daneben das nächste Byte usw. Das 33. Byte ist also das erste Zeichen in der zweiten Zeile. Insgesamt benötigt diese Pattern Descriptor Table 768 Bytes (24 Zeilen a 32 Spalten).
2. Die Color Table,also die Farbtabelle. Diese bestimmt die Farben der Zeichen jedes Charaktersatzes (jeweils 8 Zeichen, wie in CALL

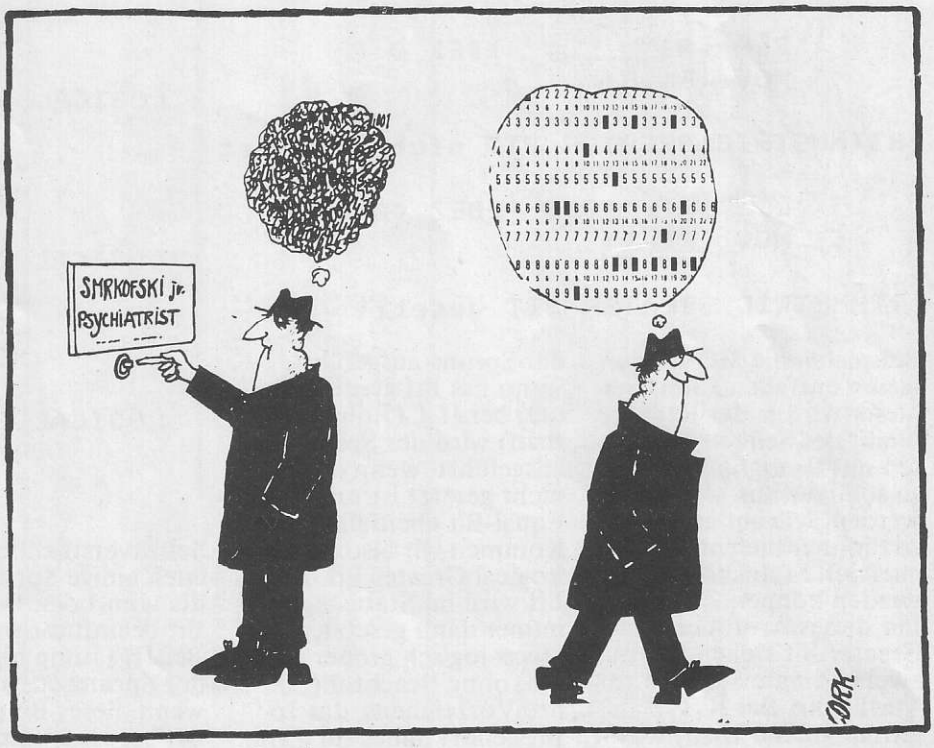
```
*****************************
* VDP-REGISTER UND TABELLEN LADEN
*

SETREG LI   1,REG          * REGISTER WERTE
REG1   MOV  *1+,Ø          * WERT AUS DER TABELLE
       JEQ  REGEND
       BLWP $VWTR          * REGISTER LADEN
       JMP  REG1
REGEND MOVB $REG+1,$>83D4 * KOPIE VDP-REG 1 FUER INTERRUPT
       LI   Ø,>Ø3ØØ        * >DØ AUF BEGINN DER SPRITE
       LI   1,>DØØØ        * ATTRIBUTE TABLE, KEINE SPRITES
       BLWP $VSBW          * WERDEN ABGEBILDET
       LI   Ø,>Ø38Ø        * COLOR TABLE
       LI   1,>FØØØ        * VORDERGRUND WEIS AUF TRANSPARENT
REG2   BLWP $VSBW
       INC  Ø
       CI   1,>Ø3AØ
       JNE  REG2
       LI   1,>Ø9ØØ        * BEGINN AB CHARAKTER >2Ø = SPACE
       MOV  1,$>834A
       BLWP $GPLLNK        * GR. BUCHSTABEN LADEN
       DATA >ØØ18
       BLWP $GPLLNK        * KL. BUCHSTABEN LADEN
       DATA >Ø04A
       LI   Ø,>Ø8FØ        * PATTERN DESCRIPTOR TABLE
       LI   1,CURSOR       * CURSOR CHAR
       LI   2,>ØØ1Ø        * 16 BYTES
       BLWP $VMBW
       RT

CURSOR DATA >3C42,>99A1,>A199,>423C * COPYRIGHT ZEICHEN
       DATA >3C3C,>2424,>2424,>3C3C * CURSOR

* DATEN FUER DIE VDP-REGISTER

REG    DATA >Ø1EØ   * REG. 1: 16KBYTE, GRAPHICS MODE
       DATA >Ø2ØØ   * REG. 2: BASIS SCREEN IMAGE TABLE
*                   * WERT MAL >4ØØ (HEX!)
       DATA >Ø3ØE   * REG. 3: BASIS COLOR TABLE
*                   * WERT MAL >4Ø (HEX!) >4Øx>ØE=>38Ø
       DATA >Ø4Ø1   * REG. 4: BASIS PATTERN DESCRIPTOR TABLE
*                   * WERT MAL >8ØØ (HEX) >8ØØx>Ø1=>8ØØ
       DATA >Ø5Ø6   * REG. 5: BASIS SPRITE ATTRIBUTE LIST
*                   * WERT MAL >8Ø (>8Øx>Ø6=>3ØØ)
       DATA >Ø6ØØ   * REG. 6: SPRITE DESCRIPTOR TABLE
*                   * WERT MAL >8ØØ
       DATA >Ø7F4   * REG. 7: FARBE FUER TEXTMODE GRAU
*                   * UND HINTERGRUND DUNKELBLAU
       DATA >ØØØØ   * KENNWERT ENDE DER TABELLE
```



TI-REVUE jeden Monat neu

COLOR im Basic, nur stimmen die Nummern der Zeichensätze nicht überein, da der Video-Display-Prozessor von 0 für die ASCII-Character 0 – 7 rechnet). Für jeden Charaktersatz brauchen wir hier ein Byte, insgesamt also 32 Bytes (256 geteilt durch 8). In jedem Byte bestimmen die ersten vier Bits (jeweils vier Bits werden auch als Nybble bezeichnet, hier ist also das höchstwertige Nybble gemeint), die Vordergrundfarbe und das niedrigwertige Nybble die Hintergrundfarbe. Auch hier gelten wieder die Werte aus dem Basic für die Farben, nur müssen wir 1 davon abziehen, also ergibt sich 0 für Transparent und >F bzw. 15 für Weiß.

3. Die Pattern Descriptor Table, die Tafel zur Bestimmung der Zeichen. Diese wird aus dem Basic mit CALL CHAR geladen und ist eigentlich für jedes Zeichen genauso aufgebaut wie der 16stellige String, den wir in Basic eingeben. Dieser String kann ja auch als 8Bytes Hexadezimal gesehen werden. Diese Tabelle umfaßt also 256 mal 8 Bytes, macht 2048 Bytes. Um Platz zu sparen, werden aber im Basic nicht alle 256 Zeichen die der Video-Prozessor darstellen kann ausgenutzt, so daß die Tabelle dann kleiner gehalten ist.

4. Die Sprite Attribute Tabelle, also die Sprite Werte Tabelle. Hier befinden sich jeweils 4 Bytes für jeden Sprite. Das erste bezeichnet die Lage in vertikaler Richtung (>FF ist ganz oben, dann folgt >00 bis >BE nach unten) und das zweite Byte bestimmt die Lage in horizontaler Richtung (>00 ist ganz links, >FF rechts). Das nächste Byte legt den Charakter-Code fest und das letzte Byte bestimmt die Farbe des Sprite im niedrigwertigen Nybble. Vom höherwertigen Nybble wird nur das letzte Bit verwendet. Ist es nicht gesetzt, ist die obere linke Ecke des Sprites mit den Angaben für die Position bestimmt und der Sprite wandert auf der rechten Seite des Bildschirmes sauber heraus. Ist dieses Bit gesetzt, dann ist die rechte obere Ecke bestimmt, d.h. der Sprite wird um 32 Pixel nach links verschoben und wandert auf der linken Seite richtig heraus. Der Video-Prozessor kann 32 Sprites verwalten, also ist diese Tabelle insgesamt 128 Bytes lang, aber auch hier wird im Extended Basic etwas Platz gespart und nur 28 Sprites werden verwendet.

5. Die Sprite Motion Table, Sprite Bewegungs-Tabelle. Diese Tabelle wird eigentlich nicht vom Video-Prozessor benötigt, sondern wird ausschließlich von der Interrupt-Routine verwaltet. Je Sprite sind auch hier wieder 4 Bytes vorhanden. Die ersten beiden geben den Geschwindigkeitswert in vertikaler und horizontaler Richtung an. Negative Zahlen geben dabei eine umgekehrte Bewegung. Die beiden weiteren Bytes werden nur von der Interrupt-Routine zur zwischenzeitlichen Speicherung von Werten benutzt.

6. Die Sprite Descriptor Table. Diese Tabelle ist genauso wie die Pattern Descriptor Table aufgebaut, nur gilt sie eben für die Sprites. Häufig wird sie auch mit der Pattern Descriptor Table zusammengelegt, so auch in Extended Basic. So brauchen wir darauf hier nicht noch weiter einzugehen.

Was liegt nun von diesen Tabellen wo im VDP-RAM. Zuerst unter TI-Basic:

VDP-Adresse HEX	DEZ	Tabellenbezeichnung
>0000	0	Screen Image Table
>02FF	767	
>0300	768	Color Table
>031F	799	
>0320	800	Pattern Descriptor Table mit Offset 96 (HEX> 60) Start daher eigentlich ab> 0000
>03EF	1007	
>03F0	1008	Start mit Charakter für ASCII 30
>07FF	2047	Ende variable je nach definierten Zeichen

Der Bereich von> 0320 bis> 03BC wird vom TI-Basic zum Umwandeln von Eingabezeilen in das Basic-Format (Crunchen) verwendet.

Nun zum Extended Basic:

VDP-Adresse HEX	DEZ	Tabellen-Bezeichnung
>0000	0	Screen Image Table
>02FF	767	
>0300	768	Sprite Attribut Table
>036F	879	
>0370	880	
>03EF	1007	Pattern Descriptor Table mit Offset 96 (HEX> 60) Start daher eigentlich ab> 0000
>03F0	1008	Start mit Charakter für ASCII 30
>077F	1919	
>0780	1920	Sprite Motion Table
>07FF	2047	
>0800	2048	Color Table
>081F	2079	

Der Bereich von >0370 bis >03EF und> 08C0 und >0967 wird vom Extended Basic zur Zwischenspeicherung von Werten und zum Umwandeln von Eingabezeilen benutzt. Das Editor-Assembler-Modul nimmt weniger Rücksicht, braucht es ja auch gemäß seiner Bestimmung nicht. Es ergibt sich folgende Aufteilung:

VDP-Adresse HEX	DEZ	Tabellen-Bezeichnung
>0000	0	Screen Image Table
>02FF	767	
>0300	768	Sprite Attribut Table
>037F	895	
>0380	896	Color Table
>039F	927	
>03A0	928	Sprite Descriptor Table nur Zeichen ab> 80 (128 dez.) bis> EF (dez 239) benutzbar.
>03FF	1023	
>0400	1024	
>077F	1919	
>0780	1920	Sprite Motion Table
>07FF	2047	
>0800	2048	Pattern Descriptor Table ohne Offset für 256 Charakter
>0FFF	4095	
>1000	4096	

Im Basic befinden sich nun oberhalb dieser Tabellen im VDP-RAM das Programm, die Werte aller Variablen und noch einige „Kleinigkeiten". Beim Editor-Assembler ist der Platz frei. Bei einem angeschlossenen Disk-Laufwerk ist ganz am oberen Ende des VDP-RAM's noch ein Speicher reserviert, in dem nichts verändert werden sollte. Zu diesem Speicherbereich werden wir, wie auch zum Aufbau eines Basicprogrammes, in einem anderen Beitrag kommen. Heute soll am Schluß ein kurzes Beispielprogramm stehen, wie alle notwendigen Tafeln des VDP-RAM's und die Register des Video-Prozessors geladen werden können. Die Kommentare geben auch Aufschluß über die Bedeutung der VDP-Register.

GRÖSSER UND KLEINER JETZT WIRD ES TRICKREICH

Jedes Modul, welches beim TI 99/4A Maschinensprache ermöglicht, stellt uns ja einige Hilfsroutinen für die Assemblerprogramme zur Verfügung, auf die wir hier zum Teil näher eingehen wollen.

Als erstes sollen die Routinen zum Beschreiben und Lesen des VDP-RAM's erwähnt werden, also VSBW, VSBR, VMBW, VMBR und VTWR. Nehmen wir die letzte Routine zuerst. Diese dient dazu, in ein bestimmtes Register des Video-Prozessors einen Wert zu schreiben. In R0 unseres Workspaces (Arbeitsbereiches) muß dabei im höherwertigen Byte die Nummer des Registers, also z.B. >01 für Register 1 und im niederwertigen Byte der Wert, der geschrieben werden soll, also z.B. >E0 stehen. Zu beachten ist hier, daß auf die Speicherstelle >83D4 immer eine Kopie des Wertes von VDP-Register 1 stehen muß. Das folgende Listing 1 zeigt, wie das gemacht werden kann:

```
* Listing 1:

        REF   VWTR  * Fuer Assembler u. MM

VWTR    EQU   >2030 * Fuer X-Basic

        LI    R0,>01E0
        BLWP  @VWTR
        LI    R0,>E000
        MOVB  R0,@>83D4

* Oder "getrickst":

        LI    R0,>01E0
WERT    EQU   $-1
        BLWP  @VWTR
        MOVB  @WERT,@>83D4
```

Der Trick im zweiten Teil des Listings bezieht sich auf eine besondere Möglichkeit mancher TMS 9900 Assembler. Das Dollarzeichen wird als Wert des augenblicklichen Standes des Programmzählers genommen. Hier wird dem Label XYZ also die Adresse des Bytes, in dem >E0 steht, zugeordnet. Ein Beispiel, wie VWTR benutzt wird, hatten wir ja schon in der letzten Ausgabe besprochen.

Die anderen Hilfsroutinen haben wir in den letzten Folgen auch schon verwendet. Dennoch wollen wir erst einmal die Abkürzungen erklären. Das V steht dabei für das VDP-RAM, S steht für Single, also Einzeln, M für Multiple, also mehrfach, B für Byte, W für Write (Schreiben) und R für Read, also lesen. VMBW steht also für mehrere Bytes in das VDP-RAM schreiben.
Die Routinen VSBW und VSBR beziehen sich also auf einzelne Bytes. Beiden gemeinsam ist, daß vor dem Aufruf R0 unser Workspace die VDP-Adresse enthalten muß, auf die wir schreiben bzw. von der wir ein Byte lesen wollen. Im höherwertigen Byte von R1 muß der Wert enthalten sein, den wir schreiben wollen, bzw. bei VSBR enthält das höherwertige Byte von R1 den Wert, den wir gelesen haben. Das Listing 2 zeigt nochmal die Routine aus der ersten Folge zum Darstellen eines A auf dem Bildschirm und danach lesen wir wieder den Inhalt der gleichen Speicherstelle.

```
* Listing 2:

        REF   VSBW,VSBR *Fuer Ass. u. MM

VSBW    EQU   >2020  * Fuer X-BAsic
VSBR    EQU   >2028  * Fuer X-BAsic

* Byte schreiben:
        LI    R0,>0190
        LI    R1,>4100  * >A100 Fuer X-Basic
        BLWP  @VSBW

* Byte lesen:
        LI    R0,>0190
        BLWP  @VSBR
* In R1 jetzt das gelesene Byte
```

TI-REVUE jeden Monat neu

```
* Listing 3:

* Fuer Assembler und Mini-Mem

        REF   VMBW,VMBR

TEXT1   TEXT 'DAS IST EIN TEST'

        LI    R0,>00A1
        LI    R1,TEXT1
        LI    R2,>0010
        BLWP  @VMBW

* Fuer X-Basic:

VMBW    EQU   >2024
VMBR    EQU   >202C

TEXT1   BYTE  >B4,>A5,>B3,>B4

        LI    R0,>00A1
        LI    R1,TEXT1
        LI    R2,>0004
        BLWP  @VMBW

* Bildschirm abspeichern:

        DEF   VSAVE,VLOAD

        REF   VMBW,VMBR * Fuer Ass. u. MM.

VMBW    EQU   >2024     * Fuer X-Basic
VMBR    EQU   >202C     * Fuer X-Basic


MYWS    BSS   >20       * Workspace reservieren

BUFFER  BSS   >300      * Speicher fuer Bildschirm


VSAVE   LWPI  MYWS
        CLR   R0        * Start VDP-RAM bei >0000
        LI    R1,BUFFER
        LI    R2,>0300
        BLWP  @VMBR

ENDE    CLR   4
        MOVB  4,@>837C  * GPL-Statusbyte loeschen
        LWPI  >83E0
        B     @>0070    * Ruecksprung

VLOAD   LWPI  MYWS
        CLR   R0
        LI    R1,BUFFER
        LI    R2,>0300
        BLWP  @VMBW
        JMP   ENDE
```

MSX®
REVUE
DAS MAGAZIN
FÜR FREUNDE
DER KOMPATIBLEN
DM 5,80/ÖS 49/SFR 5,80

NEU!

IM TEST:
Philips 8020
Spectravideo 728
Sony
Creative Graphics
Yashica 64
Philips Printer 0020
Ackobase
Ackotext
Sony Plotter C 41
Quickdisk QDM/01

LISTINGS:
32 Seiten
MSX-Programme

MARKTÜBER-
SICHT:
Das komplette
MSX-Software-
Angebot!
Alle Fachbücher!

```
* Listing 4:
        REF   KSCAN    * Fuer Ass. u. MM

KSCAN   EQU   >201C    * Fuer X-Basic
* Warten auf gedrueckte Taste:

        CLR   R4
        MOVB  R4,@>8374 * Tastaturmodus
        MOVB  R4,@>837C * GPL-Status loeschen
TASTE   BLWP  @KSCAN
        MOVB  @>837C,R4
        JEQ   TASTE

* Warten auf Leertaste

SPACE   DATA  >2000
        CLR   R4
        MOVB  R4,@>8374 * Tastaturmodus
TASTE   BLWP  @KSCAN
        MOVB  @>837C,R4
        COC   @SPACE,R4 * Bit 2 pruefen
        JNE   TASTE
        MOVB  @>8375,R4
        C     R4,@SPACE
        JNE   TASTE
```

Auch die Routinen VMBW und VMBR hatten wir schon angesprochen, so daß hier nur noch einmal kurz darauf eingegangen werden muß. Bei beiden Routinen muß R0 wieder die VDP-Adresse enthalten, R1 enthält einen Zeiger auf den Beginn des normalen RAM's, von wo aus geschrieben bzw. wohin gelesen werden soll und R2 enthält die Anzahl der Bytes. Das Listing 3 zeigt die Anwendung zum Schreiben eines Textes jeweils für das Assembler bzw. Mini-Memory Modul, wenn die Routine aus dem Basic bzw. Extended Basic aufgerufen wird und ein Beispiel dafür, wie der komplette Bildschirm in das RAM gespeichert wird.

Anzumerken zu VMBW und VMBR bleibt noch, daß niemals in R2 Null stehen darf, sonst hängt sich der Rechner bei einigen der Module auf. Hier ist wohl Texas Instruments eine kleine „Ungenauigkeit" unterlaufen. Normalerweise müßte dieser „Fehler" wohl in der Routine berücksichtigt werden.

Kommen wir nun zu einer weiteren wichtigen Hilfsroutine: KSCAN. Diese dient der Tastaturabfrage. Benötigt werden hier keine besonderen Vorbereitungen in den Registern unserer Workspace, sondern es müssen einige Speicherstellen gesetzt werden. Auf >8374 müssen wir ein Byte mit dem Wert des Tastaturmodus belegen. Dabei gelten die gleichen Werte wie beim Basic-Befehl CALL KEY. Bei dem Wert >01 wird also die linke Tastatur mit Joystick und bei >02 die rechte abgefragt. Die anderen Werte,> 00, >03, >04 und >05 sind identisch zum CALL KEY. Nach der Tastaturabfrage wird auf >8375 der ASCII-Wert der gedrückten Taste (auch hier gelten die gleichen Werte wie bei CALL KEY) und auf >8376 und >8377 gegebenenfalls die Werte für die Joysticks. Wenn eine neue Taste gedrückt wurde, ist das Bit 2 im GPL-Statusbyte (zu finden auf >837C) gesetzt, andernfalls ist das Bit (nicht das ganze Byte!) 0. Das Listing 4 zeigt zwei Anwendungen: Einmal wird nur gewartet, bis irgendeine Taste gedrückt wird und zum anderen wird solange gewartet, bis die Leertaste gedrückt wird.

Die letzte Der Hilfsroutinen soll XMLLNK sein. Diese erlaubt uns, auf einfache Art und Weise auf Routinen des Betriebssystem zuzugreifen. Je nachdem welche Routine wir benutzen wollen, müssen wir entsprechende Speicherbereiche vorbereiten. Wir wollen hier am Beispiel von CIF (Convert Integer – Floiting Point, also der Umwandlung einer Integer-Zahl in eine Fließkommazahl) den Gebrauch von XMLLNK zeigen (Listing 5). Dabei wird ein ganzes Wort, d.h. zwei Bytes in die im TI 99/4A benutzte Darstellung der Fließkomma-Zahlen umgewandelt.

```
* Listing 5:
* Fuer Assembler und Mini-Memory:

        REF   XMLLNK


        LI    R1,>0001 * Dieser Wert soll
        MOV   R1,@>834A * umgewandelt werden
        BLWP  @XMLLNK
        DATA  >2300     * Fuer Mini-Mem >7200
                        * Fehler im Handbuch

* Fuer Extended Basic:
XMLLNK  EQU   >2018
        LI    R1,>0001 * Dieser Wert soll
        MOV   R1,@>834A * umgewandelt werden
        BLWP  @XMLLNK
        DATA  >0020
```

Anzumerken zu XMLLNK bleibt noch, daß eine Veröffentlichung in einer anderen Zeitschrift hier zu etwas Unsicherheit geführt hat. Auch das XMLLNK des Extended Basic Moduls arbeitet einwandfrei, nur müssen andere DATA-Werte eingesetzt werden, wie es das Beispiel zeigt. Die Werte, wie auch sonstige Equates für das Extended Basic-Modul, finden sich in der Anleitung zum Editor-Assembler.

DER SCHLÜSSEL ZUM BETRIEBSSYSTEM

Zuerst wollen wir uns mit GPLLNK und DSRLNK beschäftigen. Diese beiden Routinen fehlen beim Extended Basic Modul.
Mittlerweile gibt es aber einige Veröffentlichungen über diese Hilfen auch bei Extended Basic, so sind auch Beispiele dafür in unserem Assembler-Sonderheft enthalten.
Wollen wir diese Routinen also in einem Assembler-Programm für Extended Basic nutzen, so müssen wir diese einfach an den Schluß des Programms (oder Anfang) anhängen und können sie dann genauso wie beim Assembler-Modul oder beim Mini-Memory aufrufen.
GPLLNK ist in seiner Wirkung und der Handhabung dem XMLLNK sehr ähnlich. Im Gegensatz zu XMLLNK, welches Routinen aus dem ROM aufruft, können mit GPLLNK Routinen, die in den GROM's der Konsole bzw. des eingesteckten Moduls enthalten sind, abgearbeitet werden. Hier stehen in der Konsole diverse mathematische Funktionen für Fließkommazahlen und Routinen zum Laden der Zeichensätze, sowie zur Ausgabe von Tönen zur Verfügung. Ein Beispiel für die Benutzung der Routinen zum Laden der Zeichensätze in den VDP hatten wir schon im Heft 9/85. Die Benutzung der im Betriebssystem vorhandenen Routinen für "Beep" und "Onk", die Töne für richtige und falsche Eingabe, finden Sie in Listing 1.
Das geht doch wirklich einfach und kann in jedes Assembler-Programm eingebunden werden, ohne großes Programmieren von Soundlisten. Eigentlich ist es doch auch nicht einzusehen, warum nicht auch in Assembler-Programmen der Benutzer akustisch auf richtige und falsche Eingaben oder das Ende des Programms u.ä. hingewiesen wird.
Bei den Ton-Routinen muß hier noch angemerkt werden, daß es einige Vorschläge für GPLLNK unter Extended Basic gibt, die hier nur einmal richtig funktionieren.Dazu gehört auch das Listing, welches TI/USA auf Anforderung verschickte. Die in unserem Assembler-Sonderheft abgedruckte Version arbeitet einwandfrei.
Kommen wir nun zum DSRLNK, und hier wird es etwas komplizierter. DSRLNK in seiner ursprünglichen Form, wie es im Betriebssystem des TI 99/4A enthalten ist, ist wohl der eigentliche Schlüssel des gesamten Betriebssystems. Darüber werden alle Haupt-Programme, Unterprogramme (auch die Basic-CALL's) und die Betriebssysteme der Peripheriegeräte wie z.B. Disk-Controller aufgerufen. Das DSRLNK für Assembler ist dagegen etwas abgemagert. Es erlaubt nur den Zugriff auf die Peripheriegeräte. Damit kann über diese Routine nicht auf den Kassettenrekorder zugegriffen werden. Das muß über GPLLNK erfolgen, wie es im Editor-Assembler-Handbuch und auch schon in der TI-REVUE kurz beschrieben wurde.
Um nun ein DSRLNK ausführen zu können, benötigen wir zuerst ein sogenannten PAB (Peripheral Access Block, was soviel heißt wie Zugriffs-Block für die Peripheriegeräte). Dieser muß wie folgt aufgebaut werden:
1. Byte: I/O Opcode (legt die Funktion fest) im Einzelnen:

> 01 OPEN
> 02 CLOSE
> 03 READ
> 04 WRITE
> 04 RESTORE
> 05 LOAD
> 06 SAVE
> 07 DELETE
> 08 SCRATCH RECORD
> 09 STATUS

OPEN, CLOSE, RESTORE, SAVE und DELETE haben die gleichen Bedeutungen wie in der Dateibehandlung im Basic, READ heißt Lesen eines Datensatzes, WRITE schreiben eines Datensatzes, LOAD ist gleichbedeutend mit OLD im Basic, SCRATCH RECORD bedeutet, daß der letzte Datensatz gelöscht werden soll und STATUS gibt Auskunft über die Form der Datei und ob irgendein Ende erreicht ist. Dies wird im Basic z.B. für die EOF-Funktion verwendet.
2. Byte: Flagbyte, enthält die wichtigen Informationen über den Typ der Datei.
Bit 0 (niedrigwertigstes Bit): Dateityp (1=Sequentiell, 0=Fixed)
Bit 1 und 2: Art der Eröffnung (00=Update, 01=Output,10=Input und 11=Append)
Bit 3: Art der Daten (0=Display, 1=Internal)
Bit 4: Art des Datensatzes (=Fixed, 1=Variable)
Bit 5 bis 7: Errorcode, werden von der DSR entsprechend gesetzt, wenn ein Fehler bei der Dateibehandlung auftrat.
Byte 3 und 4: Zeiger zur Adresse des Puffers im VDP-RAM für den Datensatz.
Byte 5: Länge des Datensatzes. Bei Fixed-Dateien steht hier die Länge, bei Variable-Dateien die maximale Länge.
Byte 6: Länge des aktuellen Datensatzes, d.h. die Länge des gerade zu schreibenden oder gelesenen Datensatzes.
Byte 7 und 8: Nummer des Datensatzes. Bei SAVE steht hier die Länge des zu speichernden Programms,

```
* Listing 1:

        REF  GPLLNK * Fuer Assembler u. MM

* Fuer X-Basic muss GPLLNK angeh{ngt
* werden

* Accept-Tone:

        BLWP @GPLLNK
        DATA >0034

* Bad-Tone:

        BLWP @GPLLNK
        DATA >0036
```

```
* Listing 2:

* PAB fuer Display-Variable 80 Datei

PABVAR BYTE >00  * OPEN
       BYTE >14  * DISPLAY-VARIABLE, INPUT
       DATA >1000 * ADRESSE BUFFER
       BYTE >50  * MAX. 80
       BYTE >50  * MAX CHARACTER COUNT
       DATA >0000 * 1. DATENSATZ
       BYTE >00  * SCREEN-OFFSET 0
       BYTE >09
       TEXT 'DSK1.TEST' * Immer diese intelligenten Namen

       EVEN

RAMBUF BSS  80  Buffer fuer den Datensatz


* Hier vorher eigenes Programm

* PAB in das VDP-RAM legen:

       LI   R0,>0900      * Willkuerliche Adresse
       LI   R1,PABVAR
       LI   R2,19         * Komplette Laenge
       BLWP @VMBW

       MOV  R0,R3         * Sichern
       AI   R3,>0009      * Zeigt nun auf Laengenbyte
       BL   @DSRAUF
       JEQ  ERROR

* Nun ersten Datensatz lesen

       LI   R0,>0900
       LI   R1,>0200      * OPCODE fuer READ
       BLWP @VSBW         * In PAB schreiben

       BL   @DSRAUF       * Datensatz lesen
       JEQ  ERROR

       LI   R0,>0905      * Zeigt auf CHARACTER COUNT
       BLWP @VSBR         * Byte lesen
       MOVB R1,R2
       SRL  R2,8          * Laenge nun in R2
       JEQ  HILF          * Bei 0 Fehlfunktion VMBR
       LI   R0,>1000      * Buffer im VDP
       LI   R1,RAMBUF     * Buffer im CPU-RAM
       BLWP @VMBR         * Datensatz nun im Buffer CPU-Ram

* Nun Datei schliessen

HILF   LI   R1,>0100      * OPCODE fuer CLOSE
       LI   R0,>0900
       BLWP @VSBW         * In den PAB schreiben
       BL   @DSRAUF       * Datei schliessen
```

```
* Das wars, das eigene Programm kann weitergehen

* Hier Errorhandling einfuegen, In RØ Errorbyte
* bei Ø DSR nicht gefunden

ERROR   B     @BEGINN


* Unterprogramm Aufruf DSR

DSRAUF  MOV   R3,@>8356
        BLWP  @DSRLNK
        DATA  >ØØØ8
        RT                    * Veraendert nicht Status
```

bei LOAD die max. Länge des Eingabebuffers, in den das Programm im VDP-RAM geladen werden soll. Ist das Programm länger, erfolgt eine Fehlermeldung.
Byte 9: Screen Offset: Wird wie im Basic für die Bildschirmdarstellung ein Screen-Offset verwendet, muß hier der Wert stehen.
Byte 10: Länge des Dateinamens. Dabei ist die gesamte Länge gemeint, also nicht nur die Länge des Namens des Peripheriegerätes.
Byte 11 folgende: Name
Diesen PAB müssen wir nun in das VDP-RAM bringen, und dann muß noch ein Zeiger auf >8356 (ganzes Wort) gelegt werden, der auf das Längenbyte des Namens im VDP-RAM zeigt. Danach kann mit dem üblichen BLWP mit nachfolgendem DATA das Betriebssystem eines Peripheriegerätes aufgerufen werden. Das folgende Listing 2 zeigt ein Beispiel dafür. Bitte beachten Sie, daß hier willkürlich in die Belegung des VDP-RAM's eingegriffen wird. Wenn Basic-Programme vorhanden sind, muß erst der entsprechende Speicherplatz für PAB und Datenbuffer reserviert werden. Dazu kommen wir ein anderes Mal.
Wenn nun ein Fehler auftritt, so ist bei der Rückkehr aus DSRLNK zum rufenden Programm das Equal-Bit im Statusregister der CPU gesetzt, d.h. durch einen einfachen Sprung, wenn gleich (JEQ), kann zu einer Routine gesprungen werden, die dann den Fehler entsprechend dem Programm behandelt.
Der letzte Hinweis für heute gilt der Eigentümlichkeit mancher Peripheriegeräte: Sie verändern die GROM-Adresse. Deshalb sollte vor dem Ausführen von DSRLNK die GROM-Adresse gelesen und danach wieder geschrieben werden. Wie das geht, zeigt das Listing 3.

```
* Listing 3:

* Grom-Adresse sichern vor DSRLNK und
* anschliessend wieder schreiben


SAVEGR DATA >ØØØØ

       MOVB @>98Ø2,@SAVEGR
       MOVB @>98Ø2,@SAVEGR+1
       DEC  @SAVEGR * Muss erniedriegt werden



       BLWP @DSRLNK
       DATA >ØØØ8



       MOVB @SAVGR,@>9CØ2
       MOVB @SAVEGR+1,@>9CØ2
```

```
↕LI
)UUUUUUUUUUU
```

PIXELWOLF

Mit dem Programm Pixelwolf kann in Extended Basic hochauflösende Grafik erzeugt werden. An Hardware benötigt man außer der Konsole das Extended Basic und die Mini-Assembler-Hardware-Erweiterung. Das Programm enthält einen Loader für das Maschinenprogramm. (Anm. d. Red.: Alle für den Mini-Assembler geschriebenen Programme lassen sich auch mit einer normalen Speichererweiterung betreiben.)

Programmbeschreibung

Die Zeilen 240 bis 300 regeln den Programmeinstieg. Hier wird entschieden, ob zunächst das Maschinenprogramm geladen wird (Menü-Punkt 1), oder ob sofort die Demonstrationsgrafik gestartet wird (Menü-Punkt 2). Zeile 340 setzt den Zeiger auf den Beginn der Namensliste für Maschinensprache-Programme und Zeile 350 trägt Programmname und Startadresse in diese Liste ein.
Die Zeilen 360–510 "Poken" das Maschinenprogramm in das Mini-Assembler-Ram. Ein mit einem Disassembler-Programm erstelltes Quellcode-Listing des Maschinenprogramms befindet sich in der Anlage. Das Maschinenprogramm belegt die Speicherplätze 24FE257F(hexadezimal) bzw. 9470–9599(dezimal).

Die Zeilen 520–540 initialisieren den Arbeitsbereich des Maschinenprogramms (2580–259F hexadezimal bzw. 9600–9631 dezimal) mit Null. Das Demo-Programm setzt dann in der Zeile 580 zunächst alle Zeichenmuster auf Leer. Zeile 600 lädt das Register 8 des Maschinenprogramms mit dem Wert dez. 128 entsprechend hex. 80, also dem Basic-Wert für das Leerzeichen.

Die Zeilen 610–710 plotten dann eine waagerechte und eine schräge Gerade sowie eine spezielle Sinuskurve auf den Bildschirm. Sie rufen dazu das Maschinenprogramm nicht direkt auf, weil die Parameterübergabe des Call-Link-Befehls in Verbindung mit dem Miniassembler-RAM nicht funktioniert. (Der Call-Link-Befehl des Extended-Basic sucht die numerischen Variablen vergeblich in der RAM-Erweiterung, während sie sich nach wie vor im VDP-RAM befinden.)

Parameterprüfung und -übergabe sowie der Maschinenprogrammaufruf werden durch eine Basic-Subroutine durchgeführt, die man in den Zeilen 760–790 findet.

Die Parameter werden direkt in das Register 0 des Maschinenprogramms geladen. Die Behandlung der Parameter und der Maschinenprogrammaufruf in der Basic-Subroutine haben den Vorteil, daß das Hauptprogramm entsprechend einfacher gestaltet werden kann. Registerbelegung im Maschinen-Unterprogramm

```
100 ! PIXELWOLF
110 !
120 ! Hochaufloesende Grafik
130 ! fuer TI99/4A
140 ! mit EXTENDED BASIC
150 ! und MINIASSEMBLER
160 !
170 ! von Wolfgang Schmitt
180 !     Uhlandstr. 10
190 !     6238 Hofheim/Ts.
200 !
210 ! Menu
220 !
230 CALL CLEAR :: ON WARNING
 NEXT
240 DISPLAY AT(10,2):"1. Mas
chinenprogramm laden"
250 DISPLAY AT(12,2):"2. Dem
o-Programm starten"
260 DISPLAY AT(16,2):"Ihre W
ahl:"
270 ACCEPT BEEP SIZE(1)VALID
ATE("12")AT(16,27):A
280 IF A=2 THEN 580
290 DISPLAY AT(20,2):"Ladevo
rgang laeuft"
300 DISPLAY AT(22,2):"Bitte
warten"
310 !
320 ! Basic-Loader
330 !
340 CALL LOAD(8196,39,248)!
Beginn der Namensliste
350 CALL LOAD(10232,80,85,78
,75,84,32,36,254)! Programmn
ame und Startadresse
360 FOR I=9470 TO 9599
370 READ P :: CALL LOAD(I,P)
380 NEXT I ! Programm laden
390 DATA 2,224,37,128,4,193,
4,194,4,195
400 DATA 4,196,4,197,4,199,2
08,192,6,195
410 DATA 6,192,209,64,6,197,
16,0,16,0
420 DATA 2,12,0,8,60,140,61,
12,193,132
430 DATA 2,12,0,32,57,140,16
1,194,192,7
440 DATA 4,32,32,40,6,193,2,
129,0,128
450 DATA 22,7,5,136,194,72,1
92,72,6,193
460 DATA 4,32,32,32,16,1,194
,65,194,137
470 DATA 2,12,0,8,58,140,162
,197,192,11
480 DATA 4,193,4,32,32,40,6,
193,6,12
490 DATA 96,204,7,67,2,12,0,
1,192,3
500 DATA 19,1,10,12,224,76,6
,193,192,11
510 DATA 4,32,32,32,2,224,13
1,224,4,91
520 FOR I=9600 TO 9632
530 CALL LOAD(I,0)
540 NEXT I ! Arbeitsbereich
 initialisieren
550 !
560 ! Demo-Programm
570 !
580 FOR I=143 TO 33 STEP -1
:: CALL CHAR(I,"0000000000000
0000"):: NEXT I
590 CALL CLEAR
600 CALL LOAD(9616,0,128)! Z
eichennamenzaehler auf Basic
-Wert fuer Leerzeichen
610 FOR X=10 TO 245 STEP 2
620 CALL PIWO(X,100)
630 NEXT X
640 FOR X=25 TO 225
650 Y=X/2
660 CALL PIWO(X,Y)
670 NEXT X
680 FOR X=20 TO 235 STEP .2
690 Y=SIN(X/20)*(100-X)+100.
5
700 CALL PIWO(X,Y)
710 NEXT X
720 GOTO 720
730 !
740 ! Routine fuer Parameter
-Uebergabe und Maschienenpro
gramm-Aufruf
750 !
760 SUB PIWO(X,Y)
770 IF X<1 OR X>256 OR Y<1 O
R Y>192 THEN 790
780 CALL LOAD(9600,X,Y):: CA
LL LINK("PUNKT")
790 SUBEND
```

TI REVUE: Die Nummer 1 in Europa!

R0	Parameterübergabe oder VDP-RAM-Adresse
R1	Von/nach VDP-RAM zu übertragendes Byte
R2	X-grob
R3	X bzw. X-fein
R4	Y-grob
R5	Y bzw. Y-fein
R6/7	Bildschirmadresse
R8	Zeichennamen-Zähler
R9	aktueller Zeichenname
R10/11	Byte-Adresse in der Zeichendefinitionsliste
R12	Bit-Muster für neuen Pixel
R13–15	nicht benutzt

Bedienung

1. Mit "FUNCTION QUIT" das Titelbild aufrufen und Extended-Basic anwählen.
2. Miniassembler-RAM einschalten.
3. Durch Eingabe von "CALL INIT <ENTER> NEW <ENTER>" werden diverse Hilfsprogramme von Extended-Basic-Modul in das Miniassembler-RAM geladen.
4. Mit "OLD CS1" das Programm Pixelwolf laden und mit "RUN" starten.
5. Beim ersten Programmstart muß aus dem Menü-Punkt 1 (Maschinenprogramm laden) angewählt werden. Später kann dann mit Punkt 2 direkt die Demografik aufgerufen werden.
6. Das Programm beendet man mit "FUNCTION CLEAR".

Wolfgang Schmitt

PALINDROM

Ein Palindrom ist eine Zahl, die von vorwärts und rückwärts gelesen den gleichen Wert darstellt, z.B. 2992. Man kann aus fast allen Zahlen durch bestimmte Additionen ein Palindrom erzeugen. Man beginnt mit einer beliebigen zwei- oder mehrstelligen Zahl, schreibt die gleiche Zahl in umgekehrter Ziffernfolge darunter und addiert beide Zahlen. Mit der Summe wird wieder so verfahren. Diese Reihe wird solange fortgesetzt, bis ein Palindrom erreicht wird. Das folgende Beispiel dient zur Illustration:

```
  194
+ 491
-----
  685
+ 586
-----
 1271
+1721
-----
 2992 Palindrom
```

Die meisten Ausgangszahlen führen sehr schnell zu einem Palindrom. Es gibt aber zwischen 100 und 200 eine Zahl, die anscheinend zu keinem Palindrom führt. Ich überlasse es dem Leser, diese Zahl ausfindig zu machen; per Hand oder mit Hilfe des Assembler-Programmes. Soweit mir bekannt ist, erreicht diese Ausgangszahl selbst nach 50 000 Additionen (das führt zu einer Zahl mit über 20 000 Stellen) noch keine Palindromform.

Zum Programm

Um das Programm nicht unnötig zu verlängern, wurde auf Dialoge verzichtet. Deshalb sind einige Erklärungen notwendig: Nach Eingabe einer Anfangszahl und 'ENTER' werden die errechneten Summen angezeigt, bis der Bildschirm gefüllt oder ein Palindrom erreicht ist. Auf Tastendruck wird die Rechnung fortgesetzt, bzw. zur Eingabe einer neuen Anfangszahl aufgefordert. Mit 'Quit' wird das Programm verlassen. Sollte die Summenzahl 28 Stellen erreicht haben ohne daß ein Palindrom erreicht wurde, wird ebenfalls nach Tastendruck zur Eingabe einer neuen Anfangszahl aufgefordert.

Der Debugger

Der Autor gehört zu den Computerfans, die versucht haben, ohne jede Vorkenntnis die Assemblerprogrammierung mit Hilfe des TI-Handbuches zu erlernen. Es war am Anfang ausgesprochen frustrierend, aber Ausdauer führte zu einem ersten Resultat.
Da Assemblerprogramme die dumme Angewohnheit haben, am Anfang nie zu laufen und sich statt dessen irgendwo aufzuhängen, gehört der Debugger zu den wichtigsten Hilfsmittel, um Fehler ausfindig zu machen.
Für diejenigen, die sich erst kurz mit dem Assembler beschäftigen oder ihn entnervt in einer Schublade aufbewahren, möchte ich anhand des Palindromprogrammes die wichtigsten Debuggerbefehle erläutern.
Nachdem Sie das Programm eingetippt haben, speichern Sie es unter DSK2.PS (Palindrom Source) ab. Bei der anschließenden Assemblierung geben Sie für das Objektfile DSK2.P0 und für das Listfile DSK2.PL ein und starten mit der Option RL. Nach erfolgter Assemblierung drukken Sie das PL-File aus. In der ersten Spalte sehen Sie die Zeilennummern und in der zweiten Spalte die relative Speicheradresse. Um zu der tatsächlichen Adresse zu kommen, müssen Sie jeweils >A000 addieren, da die LOAD und RUN Option das Object-File im High-Memory ab A000 abspeichern wird.
Wählen Sie jetzt die 'LOAD and RUN' Option und geben Sie folgendes ein: DSK2.P0, Enter, DSK1.DEBUG, Enter, Enter, DEBUG. Mit M A000,FFFF zeigt Ihnen der Rechner die Speicherbelegung ab Adresse A000. Durch Tastendruck können Sie die Anzeige unterbrechen und wieder starten, Fctn X bricht den Befehl ab. Viel werden Sie nicht erkennen, lediglich das Auftauchen des Wortes 'PALINDROM' zeigt Ihnen, daß es sich hier wirklich um Ihr Programm handelt.

Als nächstes wollen wir das Programm aus dem Debugger heraus starten. Dazu drücken wir 'R'. Als Prompt wird der derzeitige Inhalt des Workspace Pointers W angezeigt, den wir durch Eingabe von A078 auf die Startadresse unseres selbst definierten Workspace MYREG (siehe PL-Ausdruck) einstellen und durch Drücken der Leertaste zum Programm Counter P weiterschalten. Die angezeigte Adresse ändern wir nach A0E6, der Startadresse unseres Programms. Durch Drükken der Leertaste kommen wir zum Statusregister, das wir auf 0000 abändern, falls dieser Wert nicht schon angezeigt wird und beenden diesen Abschnitt mit Tastendruck.

Als nächstes setzen wir einen Breakpoint durch Eingabe vom B A10C und starten unser Programm mit E. An dem vorgewählten Breakpoint hält das Programm an und wir können mit M A000,FFFF die Auswirkung des ersten Programmabschnitts betrachten. Sie werden feststellen, daß die Blöcke für Z1, Z2 und Buffer mit Nullen bzw. mit dem ASCII-Code für Leerstellen gefüllt sind. Nach Eingabe von W können auch die Inhalte der Register R0 bis R15 inspiziert werden.

Sie können jetzt einen oder mehrere Breakpoints setzen und mit dem E Befehl abschnittsweise durch das Programm laufen.

```
       DEF  START
       REF  KSCAN,VMBW,VSBW,VMBR
*
H1     BYTE 1                  H1=1, Byte-Konstante
H1Ø    BYTE 1Ø                 H1Ø=1Ø
H48    BYTE 48                 H48=48, ASCII-Code fuer Null
       EVEN
Z1     BSS  28                 Block von 28 Byte, zu untersuchende Zahl
Z2     BSS  28                 Hilfsblock fuer Addition
BUFFER BSS  28                 Block fuer Z1 im ASCII-Code
BUF1   BSS  32                 Buffer fuer Bildschirm-Scroll
MYREG  BSS  32                 Block fuer Arbeitsregister
TITEL  TEXT '*************'    Text fuer Programmtitel
       TEXT '*           *'
       TEXT '* PALINDROM *'
       TEXT '*           *'
       TEXT '*************'
       EVEN
FRAGE  TEXT 'ANFANGSZAHL?'     Text fuer die Eingabe der Anfangszahl
*
START  LWPI MYREG              Arbeitsregister laden
*
* Titelbild
*
       LI   RØ,>48             RØ=Adresse der 3. Bildschirmzeile, 9. Spalte
       LI   R1,TITEL-13        R1=Adresse 13 Byte vor Titel
       LI   R2,13              R2=13, Anzahl der zu schreibenden Bytes
LOOP3  AI   RØ,32              RØ=RØ+32, d.h. eine Bildschirmzeile tiefer
       AI   R1,13              R1=R1+13, d.h. naechste Textzeile
       BLWP @VMBW              Textzeile auf dem Bildschirm anzeigen
       CI   RØ,>E8             RØ=>E8 ? (Bildschirmposition der 5. Textzeile)
       JNE  LOOP3              Nein -> naechste Textzeile  anzeigen
*
NEU    BL   @INIT              Zahlen initialisieren
*
* Anfangszahl eingeben
*
       LI   RØ,>2A2            22. Bildschirmzeile, 3. Spalte
       LI   R1,FRAGE           R1 mit der Adresse von 'FRAGE' laden
       LI   R2,12              12 Bytes sollen uebertragen werden
       LIMI Ø                  Interrupt unterdruecken
       BLWP @VMBW              Textzeile auf dem Bildschirm anzeigen
       LIMI 2                  Interrupt zulassen
*
       LI   RØ,>2BØ            22. Bildschirmzeile, 17. Spalte
NEXT8  BL   @SCAN              Unterprogramm SCAN aufrufen
       CI   R1,>ØDØØ           wurde ENTER (ASCII-Code >ØD) gedrueckt?
       JEQ  NEXT9              Ja -> weiter mit NEXT9
       CI   R1,>3ØØØ           ASCII-Code der gedrueckten Taste < ASCII Null ?
       JLT  NEXT8              Ja -> Annahme verweigern, noch einmal
       CI   R1,>39ØØ           ASCII-Code der gedrueckten Taste > ASCII 9 ?
       JGT  NEXT8              Ja -> Annahme verweigern, noch einmal
       INC  RØ                 Spaltenadresse erhoehen
       LIMI Ø                  Interrupt unterdruecken
       BLWP @VSBW              gedrueckte Taste auf dem Bildschirm anzeigen
       LIMI 2                  Interrupt zulassen
       JMP  NEXT8              naechste Ziffer holen
*
* Zahl vom Bildschirm nach Z1 uebertragen
*
NEXT9  LI   R2,>2BØ            ein Byte vor der ersten Ziffer
       NEG  2                  R2=-R2
       A    RØ,R2              R2=RØ+R2, d.h. R2=Anzahl der eingegebenen Ziffern
       LI   RØ,>2B1            ab hier soll vom Bildschirm gelesen werden
       LI   R1,Z1+28           R1=Adresse von Z1+28
       S    R2,R1              R1=R1-R2, d.h. R1=Adresse der 1. Ziffer in Z1
       LIMI Ø                  Interrupt unterdruecken
       BLWP @VMBR              eingegebene Zahl vom Bildschirm nach Z1 kopieren
       LIMI 2                  Interrupt zulassen
       BL   @ZAHL              Unterprogramm ZAHL aufrufen
*
* Pointer auf 1. Ziffer von Z1
*
LOOP2  LI   R4,-1              R4=-1
LOOP1  INC  R4                 R4=R4+1
       MOVB @Z1(R4),@Z1(R4)    Z1 byteweise auf sich selbst kopieren, dient nur
       JEQ  LOOP1              der Abfrage ob Ziffer von Z1 noch null
       CI   R4,Ø               Pointer = Ø, d.h. Z1 ist 28 Ziffer lang ?
       JEQ  STOP               ja -> Stop
*
* Aufruf von Unterprogrammen
*
NEXT6  BL   @STRING
       BL   @SHOW
       BL   @PALIN
*
* Bildschirm einmal gefuellt ?
*
       DEC  R6                 R6=R6-1
       JNE  NEXT               R6=Ø? Nein -> noch keine 24 Zahlen angezeigt
       BL   @SCAN              Bildschirm mit Zahlen voll, Halt, auf Tasten-
       LI   R6,24              druck warten, dann R6 wieder gleich 24 setzen
*
* Z1 nach Z2 kopieren
*
NEXT   LI   R2,28              R2=28
NEXT1  DEC  R2                 R2=R2-1
       MOVB @Z1(R2),@Z2(R2)    die Zahl wird byteweise kopiert
       C    R2,R4              sind schon alle Ziffern kopiert?
       JNE  NEXT1              Nein -> NEXT1
*
* Addition Z1 + Z2 =>Z1
*
       LI   R2,28              R2=28, max. Anzahl von Ziffern in Z1
       MOV  R4,R3              Pointer nach R3 kopieren
       DEC  R3                 R3=R3-1
NEXT2  DEC  R2                 R2=R2-1
       INC  R3                 R3=R3+1
       CLR  R5                 R5=Ø
       AB   @Z2(R3),@Z1(R2)    Z2 von vorne, Z1 von hinten ziffernweise addieren
       MOVB @Z1(R2),R5         Ziffernsumme nach R5 kopieren
       SWPB 5                  und ins LSB bringen
       CI   R5,9               groesser 9 ?
       JGT  NEXT3              ja -> NEXT9, Uebertrag durchfuehren
       C    R4,R2              schon alle Ziffern addiert ?
       JNE  NEXT2              nein -> naechste Addition durchfuehren
       JMP  LOOP2              ja -> LOOP2, Pointer neu einstellen
*
```

```
NEXT3  SB   @H10,@Z1(R2)      von der Summenziffer 10 subtrahieren
       AB   @H1,@Z1-1(R2)     zu der Ziffer davor 1 addieren
       C    R2,R4             schon alle Ziffern addiert ?
       JNE  NEXT2             nein -> naechste Addition durchfuehren
       JMP  LOOP2             ja -> LOOP2, Pointer neu einstellen
*
* naechste Anfangszahl
*
STOP   BL   @SCAN             Tastatur abfragen, falls Taste gedrueckt weiter
       LI   R1,>2000          MSB von R1 mit ASCII-Code fuer Leertaste laden
       LI   R0,>2FF           R0 = Adresse der unteren rechten Bildschirmecke
       LIMI 0                 Interrupt unterdruecken
LOOP5  BLWP @VSBW             auf dem Bildschirm anzeigen
       DEC  R0                Bildschirmadresse erniedrigen
       JNE  LOOP5             Bildschirm vollstaendig geloescht ? nein -> LOOP5
       LIMI 2                 Interrupt zulassen
       JMP  NEU               naechste Anfangszahl holen
*
* Beginn der Unterprogramme
* =========================
*
* Abfrage ob Palindrom erreicht
*
PALIN  LI   R2,28             max. Anzahl der Ziffern in Z1
       MOV  R4,R3             Pointer nach R3 kopieren
       DEC  R3                R3=R3-1
NEXT4  DEC  R2                R2=R2-1
       INC  R3                R3=R3+1
       CB   @Z1(R2),@Z1(R3)   Ziffern von Z1 gegenlaeufig vergleichen
       JEQ  NEXT5             Ziffern sind gleich ?  ja -> NEXT5
       RT                     zurueck ins Hauptprogramm
NEXT5  C    R2,R4             Alle Ziffern verglichen ?
       JEQ  STOP              ja -> STOP
       JMP  NEXT4             nein -> naechstes Ziffernpaar vergleichen
*
* Z1 auf den Bildschirm bringen
*
SHOW   LI   R2,32             R2=32, Anzahl der Positionen pro Zeile
       LI   R0,-32            R0=-32
       LI   R1,BUF1           R1 enthaelt Adresse des RAM-Buffers
       LIMI 0                 Interrupt unterdruecken
SC     AI   R0,64             Leseadresse um 64 Byte (2 Zeilen) erhoehen
       BLWP @VMBR             32 Byte vom Schirm in den RAM-Buffer kopieren
       S    R2,R0             Schreibadresse um 32 Byte (1 Zeile) erniedrigen
       BLWP @VMBW             32 Byte aus dem RAM-Buffer ins VDP-RAM kopieren
       CI   R0,>02C0          letzte Zeile kopiert ?
       JNE  SC                nein -> weiter scrollen
*
       LI   R0,>02E2          Schreibadresse = unterste Zeile, 3. Spalte
       LI   R1,BUFFER         Leseadresse = Buffer, enthaelt ASCII-Code von Z1
       LI   R2,28             28 Byte
       BLWP @VMBW             ins VDP-RAM uebertragen
       LIMI 2                 Interrupt zulassen
       RT                     zurueck ins Hauptprogramm
*
* Zahl in String unwandeln
*
STRING LI   R1,28             max. 28 Byte sind zu transformieren
NEXT7  DEC  R1                R1=R1-1
MOVB @Z1(R1),@BUFFER(R1)  Z1 nach Buffer kopieren
AB   @H48,@BUFFER(R1)  zu jeder Ziffer den ASCII-Code von Null addieren
C    R1,R4             alle Ziffern transformiert ?
       JNE  NEXT7             nein -> naechste Ziffer
       RT                     zurueck ins Hauptprogramm
*
* String in Zahl umwandeln
*
ZAHL   SB   @H48,*R1          ASCII-Code von Null subtrahieren
       INC  R1                R1=R1+1
       DEC  R2                R2=R2-1
       JGT  ZAHL              R2>0 ? ja ->  naechste Ziffer transformieren
       RT                     zurueck ins Hauptprogramm
*
* Tastatur scannen
*
SCAN   LI   R3,>2000          R3 = 0010000000000000
       CLR  R1                R1=0
       LIMI 0                 Interrupt unterdruecken
       BLWP @KSCAN            Tastatur scannen
       LIMI 2                 Interrupt zulassen
       MOV  @>837C,R2         Statusbyte nach R2 kopieren
       COC  R3,R2             Einsen vergleichen, d.h. wurde Taste gedrueckt ?
       JNE  SCAN              nein -> noch einmal scannen
       MOVB @>8375,R1         ja -> ASCII-Code der Taste nach R1 kopieren
       RT                     zurueck ins Hauptprogramm
*
* Z1,Z2 mit Nullen, Buffer mit Leerzeichen fuellen
*
INIT   LI   R6,24             Zaehler fuer einen kompletten Bildschirm-Scroll
       CLR  R1                R1=0
       LI   R2,>2020          R2 mit ASCII-Code fuer Leerzeichen laden
       LI   R3,28             28 Bytes sollen uebertragen werden
LOOP4  MOV  R1,@Z1-2(R3)      Z1 von oben mit Nullen fuellen
       MOV  R1,@Z2-2(R3)      Z2 "   "    "   "      "
       MOV  R2,@BUFFER-2(R3)  Buffer mit Leerzeichen fuellen
       DECT R3                R3 um 2 erniedrigen
       JNE  LOOP4             R3=0 ? Nein -> nach LOOP4 springen
       RT
*
       END
```

Ich hoffe, durch diese detaillierte Beschreibung allen Mut gemacht zu haben, die bis jetzt vor der Assemblerprogrammierung zurückgeschreckt sind oder frustriert wieder aufgegeben haben.

Dr. Uwe Schulze

UTILITIES FÜR INTERRUPT-GESTEUERTE SOUNDLISTEN

Dieses Programm erweitert Ihren TI 99/4A um folgende Maschinen-Routinen:

CALL LINK("SLINIT", Bytezahl) initialisiert die Soundlistenerstellung und muß vor allen anderen Sound-Routinen stehen. Es ist empfehlenswert, diesen Befehl folgendermaßen in einem LOAD-Programm unterzubringen:

```
100 CALL INIT
110 CALL LOAD
      ("DSK1.UTILITIES")
120 CALL LINK
      ("SLINIT",1000)
```

Die 1000 ist dabei die Anzahl der im VDP-RAM zu reservierenden Bytes.

CALL LINK ("LOESCH") reinitialisiert den Speicherbereich im VDP-RAM, der von den Soundlisten benutzt wird. Darauf muß ein NEW folgen.

CALL LINK ("LABEL", Label) definiert ein Label von maximal 6 Zeichen, daß von den Unterprogrammen "SPRUNG" und "XSOUND" angesprungen werden kann. Es können bis zu zehn Labels definiert werden. Bei einem Überlauf wird eine MEMORY FULL Fehlermeldung ausgegeben.

CALL LINK ("FREQ1", Frequenz) gibt die Frequenz für Tongenerator #1 an (Frequenz 110–32767 Hz).

CALL LINK (("FREQ2", Frequenz) gibt die Frequenz für Tongenerator #2 an (Frequenz 110–32767 Hz).

CALL LINK ("FREQ3", Frequenz) gibt die Frequenz für Tongenerator #3 an (Frequenz 110–32767 Hz).

CALL LINK ("FREQ4", Frequenz) gibt die Frequenz und Art (weißes Rauschen oder periodisches Rauschen) des Geräuschgenerators an (siehe Seite 86 der Bedienungsanleitung zum TI 99/4A).

CALL LINK ("LAUT1", Lautstärke) gibt die Lautstärke für Tongenerator #1 an (Lautstärke 0–30).

CALL LINK ("LAUT2", Lautstärke) gibt die Lautstärke für Tongenerator #2 an (Lautstärke 0–30).

CALL LINK ("LAUT3", Lautstärke) gibt die Lautstärke für Tongenerator #3 an (Lautstärke 0–30).

CALL LINK ("LAUT4", Lautstärke) gibt die Lautstärke des Geräuschgenerators an (Lautstärke 0–30).

CALL LINK ("DAUER", Dauer) definiert die Dauer des Tons, der zuvor durch die Unterprogramme "FREQ" und "LAUT" bestimmt wurde. Jede Frequenz- oder Lautstärkenangabe muß mit DAUER abgeschlossen werden (Dauer 0–255 in 1/50 Sekunden gemessen). Wenn als Dauer 0 angegeben wird, stoppt die interruptgesteuerte Soundlistenverarbeitung bis sie durch einen "XSOUND"-Befehl wieder neu begonnen wird.

CALL LINK ("SPRUNG", Label) Wird dieser Befehl vom Interruptprogramm gefunden, springt es zu einem vorher definiertem Label und verarbeitet die dort stehende Soundliste.

CALL LINK ("XSOUND", Label) startet die Soundlistenverarbeitung an der Stelle, an der das Label definiert wurde.

CALL LINK ("STOP") unterbricht die Soundlistenverarbeitung und löscht alle Ton- und Geräuschgeneratoren.

Mit CALL PEEK (–31794,X) kann man abfragen, ob das Ende der Soundliste bereits erreicht ist. Wenn X=0 ist, wird gerade keine Soundliste abgearbeitet. Zum besseren Verständnis der Befehle sind hier zwei kleine Demos abgedruckt. Dieses Beispiel erzeugt das Geräusch eines Schusses:

```
1 REM**********************
2 REM*                     *
3 REM*     SOUND-DEMO      *
4 REM*    ------------     *
5 REM*                     *
6 REM* 6.7.85 T. MIELKE   *
7 REM*                     *
8 REM**********************
9 REM
10 CALL LINK("SLSTOP")
11 ALL LINK("LOESCH")
12 ALL LINK("LABEL","SCHUSS")
13 CALL LINK("FREQ4",-8)
14 CALL LINK("LAUT4",0)
15 FOR I=0 TO 30
16 CALL LINK("FREQ3",4000-I*30)
17 CALL LINK("LAUT4",I)
18 CALL LINK("DAUER",1)
19 NEXT I
20 CALL LINK("LAUT4",30)
21 CALL LINK("DAUER",0)
22 CALL LINK("XSOUND","SCHUSS!■0¬
23 PRINT "BITTE TASTE DRUECKEN!": :
: : :
24 CALL KEY(0,A,B)
25 IF B<>1 THEN 24
26 CALL LINK("XSOUND","SCHUSS")
27 GOTO 24
```

Unter dem folgenden Geräusch könnte man sich den Start einer Rakete vorstellen:

Laden Sie den Object-Code, nachdem Sie den Speicher mit CALL INIT initialisiert haben. Geben Sie CALL LINK ("SLINIT",1000) und NEW ein. Danach prüfen Sie mit einer der Demos, ob das Programm funktioniert.

An dieser Stelle soll die Routine zur Reservierung von statischem Speicherplatz im VDP-RAM in Harmonie mit dem Basic-Interpreter ausführlich erklärt werden. Der Diskcontroler benutzt die Adresse >8370 (VDP-RAMTOP) als Zeiger zum Anfang der Diskbuffer.

Was liegt näher, als den benötigten VDP-RAM-Buffer selbst als Disk-Buffer zu initialisieren? Zuerst werden zur Bufferlänge vier Bytes dazugezählt. Vom VDP-RAMTOP wird jetzt die Bufferlänge abgezogen. >8370 zeigt nun auf den Beginn Ihres Disbuffers. Dort müssen Sie nur noch den Header anfügen.

Er besteht aus vier Bytes:
1. Byte: >AA Erkennungsbyte für den Disk-Controler
2. und 3. Byte: Zeiger auf den nächsten Diskbuffer
4. Byte: >00 CRU-Adresse (hier keine)

Nach diesen vier Bytes stehen Ihnen alle folgenden Bytes bis zum nächsten Diskbuffer zur Verfügung. Nach dem Aufruf dieser Routine (siehe SLINIT im Quellisting) durch CALL LINK und NEW wird der Speicherplatz im VDP-RAM für Ihre Zwecke reserviert. Ein CALL FILES sollte danach vermieden werden.

Thomas Mielke

```
*****************************************************************************
*                                                                           *
*         Interruptgesteuerte Soundlistenverarbeitung fuer BASIC            *
*         ------------------------------------------------                  *
*                                                                           *
*     Fertiggestellt am 10.09.1985                  Thomas Mielke           *
*                                                   Falkenkamp 17b          *
*     Konfiguration: TI 99, X-BASIC, 32K, Disk      2000 Norderstedt        *
*                                                                           *
*****************************************************************************


       DEF  SLINIT,LABEL,FREQ1,LAUT1,FREQ2,LAUT2,FREQ3,LAUT3
       DEF  FREQ4,LAUT4,DAUER,SPRUNG,XSOUND,STOP,LOESCH

VSBW   EQU  >2020
VMBW   EQU  >2024
XMLLNK EQU  >2018
SOUND  EQU  >8400
STRREF EQU  >2014
NUMREF EQU  >200C
ERR    EQU  >2034
RAMTOP EQU  >8370

LAENGE DATA 0            * Adresse des Laengenbytes am Anfang des Eintrags
SBYTE  DATA 0            * aktuelles Sound-Byte (bei der Listenerstellung)
SCOUNT DATA 0            * zur Berechnung der Laenge eines Listeneintrags
SBOUND DATA 0            * Obere Grenze des Bereichs fuer Soundlisten
LCOUNT DATA 0            * Anzahl der benutzten Marken
LBUFF  BSS  80           * Buffer fuer 10 Labels
       BYTE 0
LENGHT BYTE 0            * Laengenbyte fuer Marke
BUFFER BSS  6            * Buffer fuer eine Sprungmarke
H01    BYTE >01
       EVEN

* Routine vergleicht aktuelles Sound-Byte mit der oberen Grenze
COMP
       C    @SBYTE,@SBOUND
       JL   COMP1
       DECT @SBYTE
       LI   R0,>0B00
       BLWP @ERR
COMP1  B    *R10

* Routine liest ein numerisches Parameter in R0 ein
NUMBER
       CLR  R0             * Null fuer einfache Variable
       LI   R1,1           * Parameternummer
       BLWP @NUMREF        * Fliesskommazahl holen
       BLWP @XMLLNK        * Rom-Routine
       DATA >12B8          * Fliesskommazahl in ganze Zahl umwandeln
       LI   R0,>300        * >03 = Ueberlauf-Fehlercode
       CB   R0,@>8354      * Ueberlauf?
       JNE  NUMBE1         * ja, dann Sprung nach NUMBE1
       LI   R0,>1E00       * Fehlercode BAD VALUE
       BLWP @ERR           * Fehler ausgeben und zurueck ins BASIC
NUMBE1 MOV  @>834A,R0      * R0 mit Integerzahl laden
       RT                  * Ruecksprung

* Routine liest String in String-Buffer ein
STRING
       LI   R0,>2020       * zwei Leerzeichen
       LI   R1,BUFFER      * Bufferadresse
       MOV  R0,*R1+        *
       MOV  R0,*R1+        *  Buffer mit Leerzeichen fuellen
       MOV  R0,*R1+        *
       CLR  R0             * einfache Stringvariable
       LI   R1,1           * Parameternummer
       LI   R2,>600        * max. Laenge laden
       MOVB R2,@LENGHT     * Laenge schreiben
       LI   R2,LENGHT      * Adresse des Laengenbytes
       BLWP @STRREF        * String holen
       RT

* Routine ermittelt Frequenzwert
DIVIDE
       LI   R1,1           * hoeherwertiges Wort von 111860
       LI   R2,46324       * niederwertiges Wort von 111860
       DIV  R0,R1          * durch R0 teilen (Ergebnis in R1)
       RT

* Routine sucht Marke im Label-Buffer
MARKE
       LI   R0,LBUFF       * R0 mit der Adresse des Buffers laden
MARKE1 MOV  R0,R1          * R0 kopieren
       C    *R0+,@BUFFER     *
       JNE  MARKE2           *
       C    *R0+,@BUFFER+2 * Vergleich:
       JNE  MARKE2           * Wenn Label nicht gefunden, Sprung nach MARKE2
       C    *R0+,@BUFFER+4 *
       JNE  MARKE2           *
       MOV  *R0,R0         * VDP-Adresse nach R0 laden
       RT                  * Ruecksprung
MARKE2 MOV  R1,R0          * Altes R0 zurueckholen
       AI   R0,8           * 8 dazu
       MOV  @LCOUNT,R2     * Anzahl der Eintraege im Label-Buffer
       AI   R2,LBUFF       * + Adresse des Label-Buffers
       C    R0,R2          * Label-Buffer leer?
       JL   MARKE1         * Nein, dann zurueck nach MARKE1
       LI   R0,>1E00       * Fehlercode BAD VALUE
       BLWP @ERR           * Fehlermeldung ausgeben und zurueck ins BASIC

* Pseudo-Diskbuffer fuer Soundlisten im VDP errichten und
* Soundlistengenerierung initialisieren
SLINIT
       MOV  R11,R10        * Ruecksprung sichern
       BL   @NUMBER        * Anzahl der zu reservierenden Bytes in R0
       AI   R0,4           * 4 dazu
       MOV  @RAMTOP,R2     * Altes VDP-RAMTOP in R2 ablegen
       MOV  R2,@SBOUND     * Obere Grenze fuer Soundlisten
       DECT @SBOUND        * - 2
       S    R0,@RAMTOP     * Anzahl der Bytes vom VDP-RAMTOP abziehen
       MOV  @RAMTOP,R0     * Neues RAMTOP in R0
       INC  R0             * R0 = erstes Byte des neuen Bereichs im VDP-RAM
       LI   R1,>AA00       * Header fuer Disk-Buffer
       BLWP @VSBW          * schreiben
       INC  R0             * + 1
       MOV  R2,R1          * High Byte des alten RAMTOP als Pointer zum
```

```
        BLWP @VSBW          * naechsten Disk-Buffer
        INC  RØ             * + 1
        SWPB R1             * Bytes tauschen
        BLWP @VSBW          * Low-Byte ins VDP-RAM schreiben
        INC  RØ             * + 1
        CLR  R1             * Ø da keine CRU-Adresse belegt wird
        BLWP @VSBW          * schreiben
        INC  RØ             * RØ = erstes freies Byte im VDP-Buffer
SLINØ1  MOV  RØ,@SBYTE      * aktuelles Soundbyte laden
        CLR  @LCOUNT        * Markenzaehler loeschen
        CLR  @SCOUNT        * Soundbyte-Zaehler loeschen
        MOV  RØ,@LAENGE     * Adresse des Laengenbytes fuer Eintrag laden
        INC  @SBYTE         * akt. Soundbyte + 1
        B    *R1Ø           * Ruecksprung

* Soundlistengenerierung initialisieren
LOESCH
        MOV  R11,R1Ø        * Ruecksprung sichern
        MOV  @RAMTOP,RØ     * RØ mit RAMTOP laden
        AI   RØ,5           * Header fuer Diskbuffer ueberspringen
        JMP  SLINØ1         * weiter in SLINIT

* Marke setzen
LABEL
        MOV  R11,R1Ø        * Ruecksprung sichern
        BL   @STRING        * String in String-Buffer einlesen
        LI   RØ,8Ø          * Obere Grenze des Label-Buffers
        C    @LCOUNT,RØ     * Buffer voll?
        JL   LABEL1         * Nein, nach LABEL1 verzweigen
        LI   RØ,>ØBØØ       * Fehlercode MEMORY FULL
        BLWP @ERR           * Fehler ausgeben und zurueck ins BASIC
LABEL1  LI   RØ,BUFFER      * Adresse des eingelesenen Labels
        MOV  @LCOUNT,R1     * Labelzaehler in R1
        AI   R1,LBUFF       * + Adresse des Label-Buffers
        MOV  *RØ+,*R1+      *
        MOV  *RØ+,*R1+      * String aus dem String-Buffer in den Label-Buffer
        MOV  *RØ+,*R1+      *
        MOV  @SBYTE,*R1     * akt. Sound-Byte wird der Wert fuer das Label
        DEC  *R1            * Wert aufs Laengenbyte des Listeneintrags legen
        LI   RØ,8           * RØ = 8
        A    RØ,@LCOUNT     * Markenzaehler erhoehen
        B    *R1Ø           * Ruecksprung

* Routine generiert Frequenzeintrag fuer Soundgenerator 1
FREQ1
        MOV  R11,R1Ø        * Ruecksprung retten
        BL   @NUMBER        * Zahl in RØ holen
        BL   @DIVIDE        * Frequenzwert ermitteln und in R1 ablegen
        MOV  R1,R2          * R1 sichern
        SLA  R1,12          * die vier niederwertigen Frequenz-Bits isolieren
        SRL  R1,4           *
        AI   R1,>8ØØØ       * OP-Code dazu
        MOV  @SBYTE,RØ      * akt. Sound-Byte nach RØ
        BLWP @VSBW          * in Soundliste eintragen
        INC  RØ             * VDP-Adresse + 1
        MOV  R2,R1          * Frequenzwert holen
        SLA  R1,4           * die 8 hoeherwertigen Bits isolieren
        BLWP @VSBW          * und ins VDP-RAM schreiben
        INCT @SBYTE         * akt. Sound-Byte um zwei erhoehen
        INCT @SCOUNT        * Byte-Zaehler um zwei erhoehen
        B    @COMP          * Ueberpruefen ob Buffer voll und BASIC-Rueckkehr

* Routine generiert Frequenzeintrag fuer Soundgenerator 2 (siehe FREQ1)
FREQ2
        MOV  R11,R1Ø
        BL   @NUMBER
        BL   @DIVIDE
        MOV  R1,R2
        SLA  R1,12
        SRL  R1,4
        AI   R1,>AØØØ
        MOV  @SBYTE,RØ
        BLWP @VSBW
        INC  RØ
        MOV  R2,R1
        SLA  R1,4
        BLWP @VSBW
        INCT @SBYTE
        INCT @SCOUNT
        B    @COMP

* Routine generiert Frequenzeintrag fuer Soundgenerator 3 (siehe FREQ1)
FREQ3
        MOV  R11,R1Ø
        BL   @NUMBER
        BL   @DIVIDE
        MOV  R1,R2
        SLA  R1,12
        SRL  R1,4
        AI   R1,>CØØØ
        MOV  @SBYTE,RØ
        BLWP @VSBW
        INC  RØ
        MOV  R2,R1
        SLA  R1,4
        BLWP @VSBW
        INCT @SBYTE
        INCT @SCOUNT
        B    @COMP

* Routine generiert Frequenzeintrag fuer den Geraeuschgenerator
FREQ4
        MOV  R11,R1Ø        * Ruecksprung sichern
        BL   @NUMBER        * Zahl in RØ holen
        NEG  RØ             * negieren
        MOV  RØ,RØ          * auf null pruefen
        JEQ  FREQ41         * Wenn null, dann Sprung nach FREQ41
        CI   RØ,8           * mit 8 vergleichen
        JLE  FREQ42         * falls kleiner oder gleich, Sprung nach FREQ42
FREQ41  LI   RØ,>1EØØ       * Fehlermeldung BAD VALUE
        BLWP @ERR           * ausgeben und zurueck ins BASIC
FREQ42  DEC  RØ             * - 1
        SLA  RØ,8           * Low-Byte RØ nach High-Byte RØ schieben
        AI   RØ,>EØØØ       * OP-Code dazu
        MOV  RØ,R1          * nach R1 (zum schreiben)
        MOV  @SBYTE,RØ      * akt. Sound-Byte nach RØ
        BLWP @VSBW          * und in die Soundliste eintragen
        INC  @SBYTE         * akt. Sound-Byte + 1
```

```
       INC  @SCOUNT       * Byte-Zaehler + 1
       B    @COMP         * Ueberpruefen ob Buffer voll und BASIC-Rueckkehr
* Routine generiert die Lautstaerke fuer Soundgenerator 1
LAUT1
       MOV  R11,R10       * Ruecksprung retten
       BL   @NUMBER       * Zahl nach RØ laden
       CI   RØ,31         * RØ kleiner 31?
       JL   LAUT11        * Ja, dann Sprung nach LAUT11
       LI   RØ,>1EØØ      * sonst Fehlermeldung BAD VALUE
       BLWP @ERR          * und Ruecksprung ins BASIC
LAUT11 SLA  RØ,7          * Low-Byte nach High-Byte schieben (das letzte
*                         * Bit wird uebersehen)
       AI   RØ,>9ØØØ      * OP-Code dazu
       MOV  RØ,R1         * nach R1 (zum schreiben ins VDP-RAM)
       MOV  @SBYTE,RØ     * VDP-Adresse in RØ
       BLWP @VSBW         * schreiben
       INC  @SBYTE        * akt. Sound-Byte incrementieren
       INC  @SCOUNT       * Byte-Zaehler incrementieren
       B    @COMP         * Ueberpruefen ob Buffer voll und BASIC-Rueckkehr

* Routine generiert die Lautstaerke fuer Soundgenerator 2 (siehe LAUT1)
LAUT2
       MOV  R11,R10
       BL   @NUMBER
       CI   RØ,31
       JL   LAUT21
       LI   RØ,>1EØØ
       BLWP @ERR
LAUT21 SLA  RØ,7
       AI   RØ,>BØØØ
       MOV  RØ,R1
       MOV  @SBYTE,RØ
       BLWP @VSBW
       INC  @SBYTE
       INC  @SCOUNT
       B    @COMP

* Routine generiert die Lautstaerke fuer Soundgenerator 3 (siehe LAUT1)
LAUT3
       MOV  R11,R10
       BL   @NUMBER
       CI   RØ,31
       JL   LAUT31
       LI   RØ,>1EØØ
       BLWP @ERR
LAUT31 SLA  RØ,7
       AI   RØ,>DØØØ
       MOV  RØ,R1
       MOV  @SBYTE,RØ
       BLWP @VSBW
       INC  @SBYTE
       INC  @SCOUNT
       B    @COMP

* Routine generiert die Lautstaerke fuer den Geraeuschgenerator (siehe LAUT1)
LAUT4
       MOV  R11,R10
       BL   @NUMBER
       CI   RØ,31
       JL   LAUT41
       LI   RØ,>1EØØ
       BLWP @ERR
LAUT41 SLA  RØ,7
       AI   RØ,>FØØØ
       MOV  RØ,R1
       MOV  @SBYTE,RØ
       BLWP @VSBW
       INC  @SBYTE
       INC  @SCOUNT
       B    @COMP

* Routine generiert die Dauer des vorher definierten Eintrags und initialisiert
* einen neuen Soundlisteneintrag
DAUER
       MOV  R11,R10       * Ruecksprung retten
       BL   @NUMBER       * Zahl in RØ ablegen
       MOV  RØ,R1         * Zahl sichern
       SLA  R1,8          * Low-Byte nach High-Byte schieben
       MOV  @SBYTE,RØ     * akt. Sound-Byte nach RØ (fuer VDP-Zugriff)
       INC  @SBYTE        * Sound-Byte incrementieren
       BLWP @VSBW         * schreiben
       MOV  @LAENGE,RØ    * Adresse des ersten Bytes des Eintrags (Laengen-Byte)
       MOV  @SCOUNT,R1    * Byte-Zaehler nach R1
       SLA  R1,8          * und 8 Bit nach links schieben
       BLWP @VSBW         * ins VDP-RAM schreiben
       MOV  @SBYTE,@LAENGE * neues Laengenbyte erhaelt Wert des Sound-Bytes
       INC  @SBYTE        * und erhoehen
       CLR  @SCOUNT       * Byte-Zaehler loeschen
       B    @COMP         * Pruefen ob Buffer voll ist und BASIC-Ruecksprung

* Routine generiert einen Sprung-Eintrag zu einem vorher definierten Label
SPRUNG
       MOV  R11,R10       * Ruecksprung retten
       BL   @STRING       * String in String-Buffer einlesen
       BL   @MARKE        * Mit schon definierten Labels vergleichen
       MOV  RØ,R2         * Sprungadresse in RØ nach R2 kopieren
       MOV  @SBYTE,RØ     * akt. Sound-Byte in RØ (fuer VDP-Zugriff)
       DEC  RØ            * kein Laengenbyte
       CLR  R1            * null = OP-Code fuer Sprung
       BLWP @VSBW         * schreiben
       INC  RØ            * VDP-Adresse erhoehen
       MOV  R2,R1         * Sprungadresse
       BLWP @VSBW         * hoeherwertiges Byte schreiben
       INC  RØ            * VDP-Schreibadresse erhoehen
       SWPB R1            * Bytes vertauschen
       BLWP @VSBW         * und niederwertiges Byte schreiben
       INC  @SBYTE        * akt. Sound-Byte +3
       INCT @SBYTE        *
       MOV  @SBYTE,@LAENGE * Adr. des Laengenbytes erhaelt Wert des Sound-Bytes
       INC  @SBYTE        * akt. Soundbyte incrementieren
       CLR  @SCOUNT       * Byte-Zaehler loeschen
       B    @COMP         * Ueberpruefen ob Buffer voll ist und Ruecksprung

* interruptgesteuerte Soundlistenverarbeitung initialisieren
XSOUND
       MOV  R11,R10       * Ruecksprung retten
       BL   @STRING       * String in String-Buffer holen
       BL   @MARKE        * String mit Labels vergleichen, Adresse in RØ
```

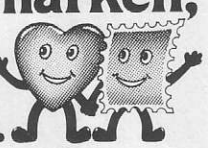
ASSEMBLER

```
       MOV  R0,@>83CC    * Adresse der Liste nach >83CC
       MOVB @H01,@>83CE  * Soundlistenverarbeitung ausloesen
       SOCB @H01,@>83FD  * VDP-RAM Flag setzen (1=VDP-RAM / 0=GROM)
       B    *R10         * Ruecksprung ins BASIC
* Stoppt die interruptgesteuerte Soundlistenverarbeitung
STOP   CLR  R0            * Soundlistenverarbeitung stoppen
       MOVB R0,@>83CE     *
       LI   R0,>9F00      * Lautstaerke Generator 1 auf 30
       MOVB R0,@SOUND     *
       LI   R0,>BF00      * Lautstaerke Generator 2 auf 30
       MOVB R0,@SOUND     *
       LI   R0,>DF00      * Lautstaerke Generator 3 auf 30
       MOVB R0,@SOUND     *
       LI   R0,>FF00      * Lautstaerke des Geraeuschgenerators auf 30
       MOVB R0,@SOUND     *
       RT                 * Ruecksprung
       END
```

```
1 REM**********************
2 REM*                    *
3 REM*     SOUND-DEMO     *
4 REM*    ------------    *
5 REM*                    *
6 REM* 6.7.85 T. MIELKE   *
7 REM*                    *
8 REM**********************
9 REM
10 CALL LINK("SLSTOP")
11 CALL LINK("LOESCH")
12 CALL LINK("LABEL","RAKETE")
13 CALL LINK("FREQ4",-8)
14 CALL LINK("LAUT4",0)
15 FOR I=400 TO 1800 STEP 10
16 CALL LINK("FREQ3",I)
17 CALL LINK("LAUT4",(I-400)/46)
18 CALL LINK("DAUER",1)
19 NEXT I
20 CALL LINK("LAUT4",30)
21 CALL LINK("DAUER",0)
22 CALL LINK("XSOUND","RAKETE")
23 PRINT "BITTE TASTE DRUECKEN!": : : : :
24 CALL KEY(0,A,B)
25 IF B<>1 THEN 24
26 CALL LINK("XSOUND","RAKETE")
27 GOTO 24
```


DANKE!

...für den Kauf von Wohlfahrtsbriefmarken, Ihrem Porto mit Herz & Verstand.



Arbeiterwohlfahrt
Deutscher Caritasverband
Deutscher Paritätischer Wohlfahrtsverband
Deutsches Rotes Kreuz
Diakonisches Werk der EKD
Zentralwohlfahrtsstelle der Juden in Deutschland

DELETE

Mit diesem Programm ist es möglich, innerhalb von nicht 'mal einer Sekunde beliebige Programmblökke aus einem XBasic-Programm zu löschen!
Nach dem Laden durch >CALL INIT >CALL LOAD ("DSK1.$X DELETE") erfolgt der Aufruf mit CALL LINK ("DELETE",START-LINE,ENDLINE) aus dem XBasic-Direktmodus!
STARTLINE und END-LINE können beliebige gültige Zeilennummern sein, sie brauchen im Programm nicht zu existieren.
Im Gegensatz zu anderen Programmen bleiben alle LIST und EDIT Funktionen voll erhalten, und das Programm ist nach dem Aufruf voll funktionsfähig, d.h. es kann gestartet, gelistet, gespeichert und editiert werden.
Allerdings wird keine Speicherplatzoptimierung durchgeführt. Wird diese gewünscht, so ist das Programm als MERGE-Datei abzuspeichern, der Speicher zu löschen und das Programm via MERGE wieder einzuladen.
Notwendige Gerätekonfiguration: TI 99/4A-Konsole, XBasic, 32 K RAM, mind. 1 Laufwerk. Zur Eingabe des Quellcodes zusätzlich: Editor/Assembler.

Peter Ulbrich

```
*************************************************************************
*                                                                       *
*    DELETE -- Loescht Programmsegmente                                 *
*                                                                       *
*    TMS-9900 ASSEMBLER, Version XBASIC                                 *
*                                                                       *
*    Programmname: @XDELETE / @XDELETES                                 *
*                                                                       *
*    1985-06-18 by Peter Ulbrich                                        *
*                                                                       *
*    Aufruf aus XBASIC-Direktmodus mit:                                 *
*    CALL LINK("DELETE",STARTLINE,ENDLINE)                              *
*                                                                       *
*    Das Assembler-Programm loescht den angegebenen Block aus dem       *
*    XBASIC-Programm. Ist STARTLINE groesser als ENDLINE, erfolgt eine  *
*    Fehlermeldung. Das XBASIC-Programm ist nach dem Aufruf voll funk-  *
*    tionsfaehig, nur die geloeschten Zeilen stehen eben nicht mehr zur *
*    Verfuegung. Allerdings wird keine Speicheplatzoptimierung vorge-   *
*    nommen. Wird diese gewuenscht, so ist folgendermassen vorzugehen:  *
*    > SAVE DSK1.Name,MERGE      )  Dabei bedeutet                      *
*    > NEW                       )  Name                                *
*    > MERGE DSK1.Name           )  irgendein gueltiger Dateiname       *
*                                                                       *
*************************************************************************



       DEF DELETE

* XBASIC-Equates

NUMREF EQU  >200C        Uebernimmt numerische Parameter
XMLLNK EQU  >2018        ROM-Routine:
CFI    EQU  >12B8        Floating Point --> Integer
ERR    EQU  >2034        Fehlerbehandlung:
ERRBA  EQU  >1C00        Bad Argument
ERRIAL EQU  >1F00        Incorrect Argument List

* Sonstige Equates

ARGNMR EQU  >8312        Argumentenanzahl
FAC    EQU  >834A        Floating Point Accumulator
LINBEG EQU  >8330        Beginn der Zeilennummerntabelle
LINEND EQU  >8332        Ende der Zeilennummerntabelle
NEXT   EQU  >0070        Next
GPLWS  EQU  >83E0        GPL-Workspace
STATUS EQU  >837C        GPL Status Byte

* Workspace, Konstante

WS     BSS  >20          Eigene Arbeitsregister
TWO    DATA 2


       EVEN

* Unterprogramme
* --------------
```

```
* Error - Routine

ERR1   LI   R0,ERRBA     Fehlerbehandlung
       JMP  ERROR
ERR2   LI   R0,ERRIAL    Fehlerbehandlung
ERROR  BLWP @ERR         Fehlermeldung ausgeben und Programm stoppen

* Parameter Uebernahme

GETPAR MOV  R11,R10           Ruecksprung sichern
       CB   @ARGNMR,@TWO+1    Argumentenanzahlkontrolle
       JNE  ERR2              Anzahl<>2 --> Error
       CLR  R0
       LI   R1,1              Zeiger auf 1. Parameter
       BLWP @NUMREF           1. Parameter holen (STARTLINE)
       BLWP @XMLLNK
       DATA CFI               und in Integer verwandeln
       MOV  @FAC,R8           1. Parameter ist nun in R8
       CLR  R0
       LI   R1,2              Zeiger auf 2.Parameter
       BLWP @NUMREF           2. Parameter holen  (ENDLINE)
       BLWP @XMLLNK
       DATA CFI               und in Integer verwandeln
       MOV  @FAC,R9           2. Parameter ist nun in R9
       C    R8,R9             Vergleich
       JGT  ERR1              1. Parameter ist groesser --> Fehler
       B    *R10              Zurueck zum Aufrufer

* Basic-Return

BSCRTN CLR  R0
       MOVB R0,@STATUS   Statusbyte loeschen
       LWPI GPLWS        GPL Arbeitsregister laden
       B    @NEXT        Zurueck nach Basic

* Hauptprogramm
* =============

DELETE LWPI WS           Eigene Arbeitsregister laden
       BL   @GETPAR      Parameter holen
       CLR  R10          Counter fuer Anzahl der zu loeschenden Bytes
       CLR  R5           Flag, ob das gesamte Programm zu loeschen ist
       MOV  @LINEND,R0   Ende der Zeilennummerntabelle holen
       DEC  R0
LOOP1  DECT R0           Aktueller Pointer auf Zeilennummer
       C    R0,@LINBEG   Vergeleich mit Beginn
       JLT  RETURN       Kleiner, also nichts zu loeschen --> XBASIC
       CLR  R1           )
       MOVB *R0,R1       ) nun die Zeilennummer
       INC  R0           ) byteweise
       CLR  R2           ) holen
       MOVB *R0,R2       )
       DEC  R0           )
       SRL  R2,8         )
       A    R2,R1        ) liegt jetzt in R1
       C    R8,R1        Vergleich mit STARTLINE
       JEQ  CONT1        Gleich --> Es ist etwas zu loeschen --> CONT1
       JLT  L1           Kleiner --> Erst pruefen: Nach L1
       DECT R0           Pointer erniedrigen
```

```
      INC   R5                Flag setzen! ( Programm wird nicht ganz geloescht)
      JMP   LOOP1             und Schleife durchlaufen
      C     R9,R1             Vergleich von ENDLINE mit der aktuellen
L1                            Zeilennummuer
      JLT   RETURN            Kleiner, also gibt es nichts zu loeschen -->
                              XBASIC
CONT1 NOP                     Weiter
      DECT  R0                Pointer erniedrigen
      AI    R10,4             Bytezaehler erhoehen
LOOP2 DECT  R0                Aktueller Pointer auf Zeilennummer
      C     R0,@LINBEG        Mit Beginn vergleichen
      JLT   CONT2             Kleiner --> CONT2 mit Pruefung
      CLR   R1                )
      MOVB  *R0,R1            )  Zeilennummer byteweise holen
      INC   R0                )
      CLR   R2                )
      MOVB  *R0,R2            )
      DEC   R0                )
      SRL   R2,8              )
      A     R2,R1             ) liegt jetzt in R1
      C     R9,R1             Vergleich ENDLINE mit aktueller Zeilennummer
      JLT   CONT3             Kleiner --> CONT3
      DECT  R0                Pointer erniedrigen
      AI    R10,4             Bytezaehler erhoehen
      JMP   LOOP2             und LOOP2 durchlaufen
CONT2 MOV   R5,R5             Pruefen, ob Flag gesetzt
      JEQ   CONT5             Nein, gesamtes Programm loeschen --> CONT5
CONT3 MOV   R0,R7             Hier wird nur ein Teil geloescht, also R0 kopieren
      A     R10,R7            R7 um R10 erhoehen
LOOP3 LI    R6,4              Schleifencounter laden
L2    MOVB  *R0+,*R7+         ) 4 Bytes von der aktuellen Adresse
      DEC   R6                ) um den Wert von R10
      MOV   R6,R6             ) verschieben
      JNE   L2                )
      AI    R0,-8             Adressen erniedrigen
      AI    R7,-8
      C     R0,@LINBEG        Schon am Beginn?
      JLT   CONT4             Ja, weiter bei CONT4
      JMP   LOOP3             Nein, LOOP3 durchlaufen
CONT4 MOV   @LINBEG,R5        Beginn holen
      A     R10,R5            um den Wert von R10 hochsetzten
      MOV   R5,@LINBEG        und wieder speichern
      JMP   RETURN            Fertig, jetzt zurueck!
CONT5 MOV   @LINEND,R5        Gesamtes Programm loeschen, dazu Ende holen
      MOV   R5,@LINBEG        und auf Beginn legen
RETURN BL   @BSCRTN           Basicreturn

      END
```

BILD-SCHIRM SPEICHERN

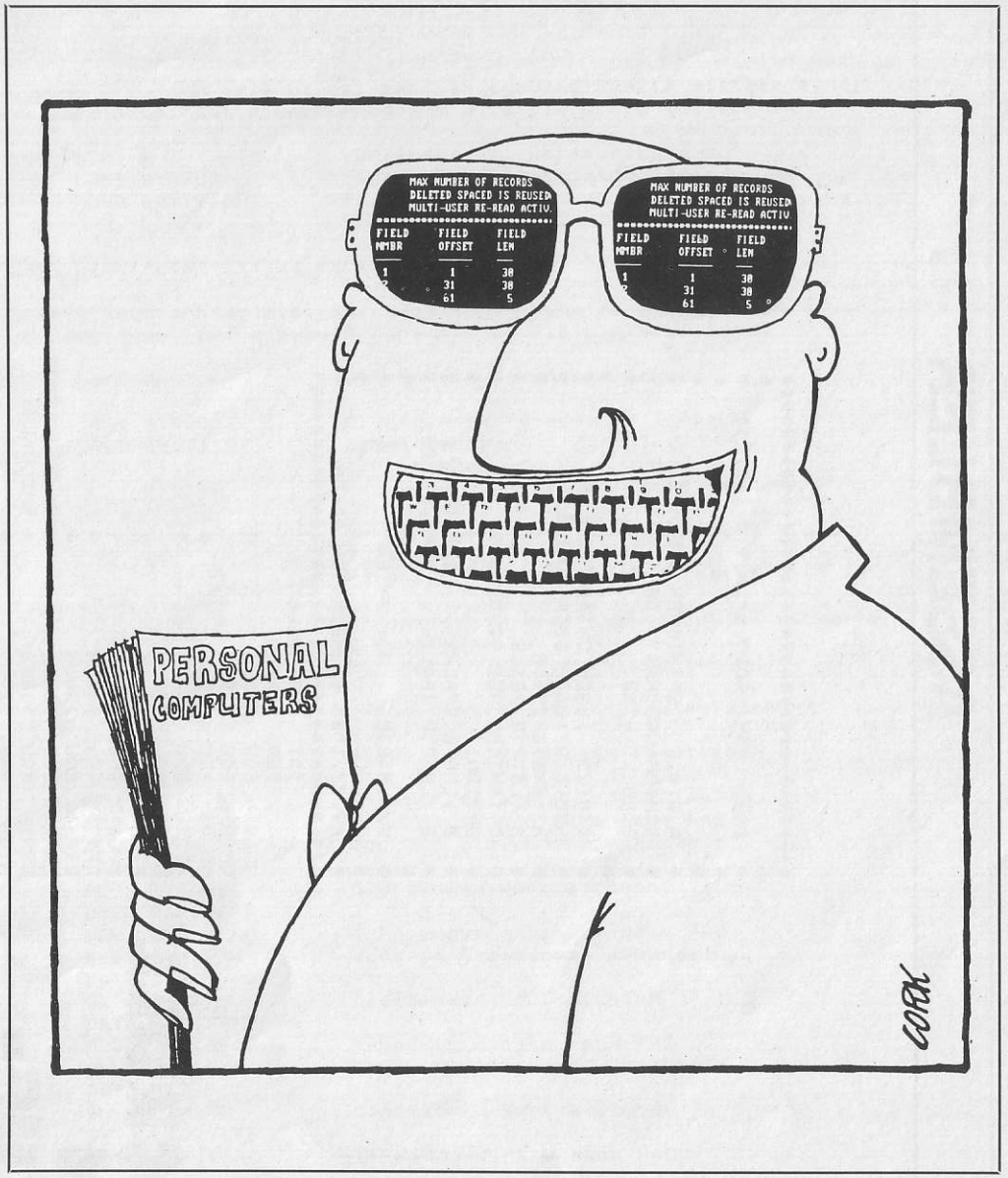
Um das Programm in TI-Basic laufen zu lassen, genügt es, nicht die Equates zu ändern, da das VDP-RAM in TI-Basic anders genutzt wird als in XBas. Call Link ("COPY",X) speichert den Bildschirm einschließlich Farben (außer Bildschirmfarbe), Sprites, Motions und Magnify. "LCOPY" holt ein abgespeichertes Bild vom Speicher und bringt es auf den Bildschirm. Der Parameter X gibt spezifiziert den Speicher. X=1:RAMspeicherbereich 1, X=2:RAMspeicherbereich 2, X=Device-Name. CALL LINK("COPY", "DSK1.BILD1") speichert also den Bildschirm als File mit Namen "BILD1" auf Diskette (10 Sektoren) ab. Leider habe ich keine Möglichkeit gefunden, die Bildschirmfarbe irgendwo auszulesen. Da beim Einlesen auch der Bereich >0370–>03EF im VDP-RAM geändert wird, ist es möglich, daß nach Drücken von "REDO" ein anderer Inhalt vorhanden ist. Die DSRLNK-Routine ist fast identisch mit der des Ed/Ass. Ich habe sie „von Hand" disassembliert und leicht geändert. Der Device-Name wird in den Bereich ab >24EA kopiert. Dort sind noch ein paar Bytes frei, da die Utilities bei >24E9 aufhören und der Loader-Pointer (>2002) ohne Maschinenprogramm schon auf >24FA steht.

Heinrich Acker

```
****************************************
*                                      *
*  CALL LINK (COPY,X)  => ABSPEICHERN  *
*                                      *
*  CALL LINK (LCOPY,X) => SCHREIBEN    *
*                                      *
*    X=1 :BILD 1        X=2 :BILD 2    *
*                                      *
*    X=STRING :DEVICE NAME             *
*                                      *
*   4750 BYTES                         *
*                                      *
*   !!!  NUR FUER EXT.BASIC  !!!       *
*                                      *
*  12.08.1985 BY HEINRICH ACKER    V1  *
****************************************
       IDT  'XCOPY10'
       DEF  COPY,LCOPY
VMBW   EQU  >2024        *
VSBW   EQU  >2020        *
VMBR   EQU  >202C        *
VSBR   EQU  >2028        *
VWTR   EQU  >2030        *
GPLWS  EQU  >83E0        *
STATUS EQU  >837C        *
FAC    EQU  >834A        *
NUMREF EQU  >200C        *
STRREF EQU  >2014        *
XMLLNK EQU  >2018        *
CFI    EQU  >12B8        *
PAD    EQU  >8300        * ARGUMENT BEZEICHNER LISTE
ERR    EQU  >2034        *
ERRSYN EQU  >0300        * SYNTAX ERROR
ERRBV  EQU  >1E00        * BAD VALUE
ERRIAL EQU  >1F00        * INCORRECT ARGUMENT LIST
ERRIO  EQU  >2400        * I/O ERROR
BUFSCR BSS  1952         * BUFFER FUER BILD 1
       DATA >E01C        * UND VDPR 1 >83D4 (MAGNIFY) UND >837A (MOTION)
       BSS  1952         * BUFFER FUER BILD 2
       DATA >E01C        * UND VDPR 1 >83D4 (MAGNIFY) UND >837A (MOTION)
PAB    EQU  >0900        * ANFANG DES PABS IM VDP
PDATA  DATA >0018,>0000,>DCDC,>0000     * VARIABLE 220,INTERNAL
       BYTE >00          *
BUFSTR BSS  >29          * PARAMETER STR-BUFFER (41 BYTES,1.BYTE=LAENGE)
STRLAN BYTE >28          * MAX. STRING-LAENGE (40 BYTES)
       EVEN              *
MYWS   BSS  >20          *
*********************************************************************
*      S P E I C H E R N      *
*******************************
COPY   LWPI MYWS
       BL   @GETVAL
       CI   R0,>0000     * PRUEFE OB BILD IN MEMORY ODER DEVICE
       JEQ  COPYD
COPYM  CLR  R0
       LI   R2,880
       BLWP @VMBR
       LI   R0,1008
       AI   R1,880
       LI   R2,1072
       BLWP @VMBR
       AI   R1,1072
       MOVB @>83D4,*R1+  * MAGNIFY ABSPEICHERN
       MOVB @>837A,*R1   * ANZAHL DER BEWEGTEN SPRITES ABSPEICHERN
       B    @ENDE
COPYD  LI   R0,>0820
       MOVB @>83D4,R1
       BLWP @VSBW        * MAGNIFY IN VDP >0820 ABSPEICHERN
       INC  R0
       MOVB @>837A,R1
       BLWP @VSBW        * ANZAHL DER BEW.SPRITES IN VDP >0821 ABSPEICHERN
       LI   R4,>0300     * R4 (HB) : WRITE OP-CODE
       LI   R1,>1A00     * OUTPUT-CODE
       BL   @CPAB
       B    @ENDE
*********************************************************************
*        L A D E N          *
*****************************
LCOPY  LWPI MYWS
       BL   @GETVAL
       CI   R0,>0000     * PRUEFE OB BILD IN MEMORY ODER DEVICE
       JEQ  LCOPYD
LCOPYM CLR  R0
       LI   R2,880
       BLWP @VMBW
       LI   R0,1008
       AI   R1,880
       LI   R2,1072
       BLWP @VMBW
       AI   R1,1072
       MOVB *R1,@>83D4   * MAGNIFY IN >83D4
       LI   R0,1
       MOVB *R1+,R0
       SWPB R0
       BLWP @VWTR        * UND VDP-REG 1 SCHREIBEN
       MOVB *R1,@>837A   * ANZAHL DER BEWEGTEN SPRITES SCHREIBEN
       B    @ENDE
LCOPYD LI   R4,>0200     * R4 (HB) : READ OP-CODE
       LI   R1,>1C00     * INPUT-CODE
       BL   @CPAB
       LI   R0,>0820
       BLWP @VSBR
       MOVB R1,@>83D4    * MAGNIFY IN >83D4
       LI   R0,1
       MOVB R1,R0
       SWPB R0
       BLWP @VWTR        * UND VDP-REG 1 SCHREIBEN
       LI   R0,>0821
       BLWP @VSBR
       MOVB R1,@>837A    * ANZAHL DER BEWEGTEN SPRITES SCHREIBEN
       B    @ENDE
*********************************************************************
*      S U B S                *
*******************************
GETVAL CLR  R0
       CLR  R1
       MOVB @>8312,R1    * >8312 ENTHAELT ANZAHL DER PARAMETER
       SWPB R1
```

```
        CI   R1,>0001      * PRUEFE OB 1 PARAMETER
        JEQ  CHBEZ         * FALLS JA --> CHeck BEZeichner
        LI   R0,ERRSYN
        BLWP @ERR          * SONST 'SYNTAX ERROR'
CHBEZ   CLR  R2
        MOVB @PAD,R2       * >8300 ENTHAELT BEZEICHNER DES 1.PARAMETERS
        CI   R2,>0000      * PRUEFE OB NUM.AUSDRUCK
        JEQ  NUREF
        CI   R2,>0200      * PRUEFE OB NUM.VARIABLE
        JEQ  NUREF
        CI   R2,>0100      * PRUEFE OB STR.AUSDRUCK
        JEQ  STREF
        CI   R2,>0300      * PRUEFE OB STR.VARIABLE
        JEQ  STREF
        LI   R0,ERRIAL
        BLWP @ERR          * SONST 'INCORRECT ARGUMENT LIST'
NUREF   BLWP @NUMREF       * BRINGT NUM.PARAMETER INS FAC
        BLWP @XMLLNK       * WANDELT FAC IN GANZE ZAHL UM
        DATA CFI
        LI   R1,BUFSCR     * BUFFER FUER BILD 1 IN R1
        MOV  @FAC,R0       * PARAWERT IN R0
        CI   R0,>0001      * PRUEFE OB PARAWERT=1
        JEQ  RETUR
        AI   R1,1954       * BUFFER FUER BILD 2 IN R1
        CI   R0,>0002      * PRUEFE OB PARAWERT=2
        JEQ  RETUR
        LI   R0,ERRBV
        BLWP @ERR          * SONST 'BAD VALUE'
STREF   MOVB @STRLAN,@BUFSTR  * MAX. STR-LAENGE IN BUFSTR
        LI   R2,BUFSTR
        BLWP @STRREF       * STRING-PARAMETER IN BUFSTR (1.BYTE=STR-LAENGE)
RETUR   RT
CPAB    MOVB R1,@PDATA+1   * INPUT/OUTPUT-CODE IN PAB-DATA
        LI   R0,PAB
        LI   R1,PDATA
        LI   R2,50
        BLWP @VMBW         * PDATA INS PAB IM VDP
        LI   R3,PAB+9
        MOV  R3,@>8356     * >8356 MUSS VDP-ADR, WO NAMENSL.STEHT  ENTHALTEN
        BLWP @DSRLNK       * DATEI OEFFNEN
        DATA 8
        JEQ  ERIO          * FALLS EQ-BIT=1 --> I/OERR
        MOVB R4,R1         * R1 (HB) : READ BZW WRITE OP-CODE
        BLWP @VSBW         * INS PAB
        CLR  R1            * R1 : AKTUELLE VDP-ADR.,AB DER KOPIERT WIRD
        CLR  R4            * R4 : SCHLEIFENZAEHLER
CPAB1   MOV  R3,@>8356
        BLWP @DSRLNK       * DATEN LESEN/SCHREIBEN
        DATA 8
        JEQ  ERIO
        INC  R4            * SCHLEIFENZAEHLER ERHOEHEN
        CI   R4,4
        JEQ  CPAB3
        CI   R4,8
        JEQ  CPAB4
        CI   R4,9
        JEQ  CPAB5
        AI   R1,220        * VDP-ADR. UM 220 ERHOEHEN
CPAB2   LI   R0,PAB+2
        BLWP @VSBW         * NEUE VDP-ADR. INS PAB
        INC  R0
        SWPB R1
        BLWP @VSBW
        SWPB R1
        JMP  CPAB1
CPAB3   LI   R1,1008
        JMP  CPAB2
CPAB4   LI   R0,PAB+5
        LI   R1,>C200      * R1 (HB) : 194
        BLWP @VSBW
        LI   R1,1888
        JMP  CPAB2
CPAB5   LI   R0,PAB
        LI   R1,>0100      * R1 (HB) : CLOSE OP-CODE
        BLWP @VSBW         * INS PAB
        MOV  R3,@>8356
        BLWP @DSRLNK       * DATEI SCHLIESSEN
        DATA 8
        JNE  RETUR         * FALLS EQ-BIT=0 (KEIN FEHLER) --> RETUR
ERIO    LI   R0,ERRIO
        BLWP @ERR          * SONST 'I/O ERROR'
ENDE    CLR  R0
        MOVB R0,@STATUS
        LWPI GPLWS
        B    @>0070        * RUECKSPRUNG INS XBASIC
*****************************************************************************
*       DSR-ROUTINE FUER XBAS.     *                                  *      *
*       ----------------------     *    R13 : ALT WP                  *      *
*                                  *    R14 : ALT PC                  *      *
* BEI ERROR :                      *    R15 : STATUS                  *      *
* EQ-BIT IM STATUS-REG. GESETZT    *                                  *      *
* R0 (HB) : FEHLERCODE (0-7)       **********************************      *
* >8304 : VDP-ADR,WO PAB ANFANG    *                                  *      *
*         MINUS >4  (OFFSET)       *    >000D : >AA     (HEADER)      *      *
*                                  *    >03B4 : >2000                 *      *
* KEIN ERROR :                     *    >0CF7 : >2E     (ASC".")      *      *
* EQ-BIT NICHT GESETZT             *                                  *      *
*                                  *                                  *      *
*****************************************************************************
DSRWS   BSS  >20
DSRLNK  DATA DSRWS,$+2     * WP : DSRWS , PC : MOMENTANER PC + 2
        MOV  *R14+,R5      * SICHERE 8 IN R5 (DATA 8  AUS DSRLNK-AUFRUF)
        SZCB @>03B4,R15    * LOESCHT EQ-BIT IN 'STATUS'-REG AUF JEDEN FALL
        MOV  @>8356,R0     * R0 : VDP-ADR,WO FILE-DESCRIPTOR LAENGE STEHT
        MOV  R0,R9
        AI   R9,>FFF8      * R9 : VDP-ADR,WO I/O FLAG STEHT
        BLWP @>2028        * (VSBR)
        MOVB R1,R3         * R1,R3 (HB) : FILE-DESCRIPTOR LAENGE
        SRL  R3,8          * R3 : FILE-DESCRIPTOR LAENGE
        SETO R4            * R4 : >FFFF
        LI   R2,>24EA      * STARTADR. FUER FILE-DESCRIPTOR KOPIE
DSR1    INC  R0            * DSR1 KOPIERT AUS VDP DEVICE-NAME IN >24EA+
        INC  R4            *
        CI   R4,>0007      * PRUEFE OB DEVICE-NAME >7
        JGT  IOERR2        * FALLS JA --> IOERR2
        C    R4,R3         * PRUEFT OB FILE-DESCR. VOLLSTAENDIG
        JEQ  DSR2          * JA --> DSR2
        BLWP @>2028        * ANDERNFALLS WEITERLESEN (VSBR)
```

```
        MOVB R1,*R2+        * KOPIERBEFEHL
        CB   R1,@>ØCF7      * PRUEFE OB BYTE="."
        JNE  DSR1           * NEIN --> DSR1
DSR2    MOV  R4,R4          * R4 : DEVICE-NAME LAENGE = LAENGE OHNE EVENT."."
        JEQ  IOERR2         * FALLS R4=Ø --> IOERR2
        CLR  @>83DØ
        MOV  R4,@>8354      * >8354 : DEVICE-NAME LAENGE
        INC  R4             * R4 : DEVICE-NAME LAENGE + 1
        A    R4,@>8356      * >8356 : ZEIGT AUF VDP-ADR. HINTER LETZTES BYTE
*                           * VON DEVICE-NAME (EVENT.WO "." STEHT)
DSR3    LWPI >83EØ          **** G P L - W O R K S P A C E ****
        CLR  R1
        LI   R12,>ØFØØ      * R12 : >ØFØØ   CRU-BASIS ADR.
DSR4    MOV  R12,R12
        JEQ  DSR5           * FALLS R12=Ø --> DSR5
        SBZ  Ø              * SCHALTE CRU (WIEDER) AUS
DSR5    AI   R12,>Ø1ØØ      * CRU-BASIS UM >Ø1ØØ ERHOEHEN
        CLR  @>83DØ
        CI   R12,>2ØØØ      * PRUEFE OB GESAMTE CRU-ADR. SCHON ABGEKLAPPERT
        JEQ  IOERR1         * FALLS JA --> IOERR1
        MOV  R12,@>83DØ     * >83DØ : CRU-BASIS ADR.
        SBO  Ø              * SONST SCHALTE CRU EIN
        LI   R2,>4ØØØ
        CB   *R2,@>ØØØD     * PRUEFE OB AN >4ØØØ >AA STEHT (HEADER)
        JNE  DSR4           * FALLS NEIN --> DSR4
        A    @DSRWS+1Ø,R2   * ALT R5+R2=    R2 : >4ØØ8  BEI (DATA 8)
        JMP  DSR7           * >4ØØ8 : POINTER AUF NAMENSTABELLE IM PERI-SYS.
DSR6    MOV  @>83D2,R2      * R2 : WEITERE POINTER
        SBO  Ø              * CRU EINSCHALTEN
DSR7    MOV  *R2,R2         * R2 : (WEITERE) POINTER (BEIM 1.MAL VON >4ØØ8)
        JEQ  DSR4           * FALLS R2=Ø --> DSR4
        MOV  R2,@>83D2      * >83D2 : (WEITERE) POINTER DES PERI-SYS
        INCT R2             * R2 : POINTER + 2
        MOV  *R2+,R9        * R9 : @(POINTER+2) = STARTADR. DES PERI-SYS
        MOVB @>8355,R5      * R5 (HB): DEVICE-NAME LAENGE
        JEQ  DSR9           * FALLS R5=Ø --> DSR9
        CB   R5,*R2+        * PRUEFE OB DEVICE-NAME LAENGE STIMMT
        JNE  DSR6           * FALLS NEIN --> DSR6
        SRL  R5,8           * R5 (LB): DEVICE-NAME LAENGE
        LI   R6,>24EA       * R6 : >24EA
DSR8    CB   *R6+,*R2+      * PRUEFE OB DEVICE-NAME STIMMT (BYTEWEISE VERGL.)
        JNE  DSR6           * FALLS IRGENDEIN BYTE NICHT STIMMT --> DSR6
        DEC  R5             * R5 : NOCH ZU PRUEFENDE BYTES
        JNE  DSR8           * FALLS NOCH NICHT ALLE GEPRUEFT --> DSR8
DSR9    INC  R1
        BL   *R9            * SPRINGE INS PERIPHERIE-BETRIEBSSYSTEM
        JMP  DSR6           * PERI-SYS LAESST BEFEHL UEBERSPRINGEN
        SBZ  Ø              * CRU AUSSCHALTEN
        LWPI DSRWS          **** D S R L N K - W O R K S P A C E ****
        MOV  R9,RØ          * RØ : VDP-ADR.,WO I/O FLAG STEHT
        BLWP @>2Ø28         * I/O FLAG LESEN  (VSBR)
        SRL  R1,13          * R1 : I/O FEHLER-CODE
        JNE  IOERR3         * FALLS R1<>Ø --> IOERR3
        RTWP
IOERR1  LWPI DSRWS
IOERR2  CLR  R1
IOERR3  SWPB R1             * R1 (HB) : FEHLERCODE
        MOVB R1,*R13        * *R13 IST RØ VON HAUPTPROGRAMM-WS
        SOCB @>Ø3B4,R15     * SETZT EQ-BIT IN 'STATUS'-REG AUF JEDEN FALL
        AI   R9,>FFFB       * R9 : PAB-ANFANG MINUS >4 IM VDP
        MOV  R9,@>83Ø4      *
        RTWP
********************************************************************************
        END
```



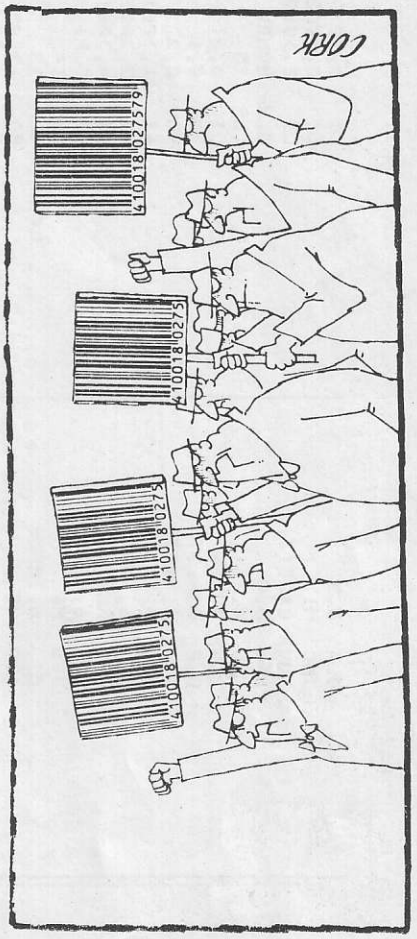
SCROLL IN AUS-SCHNIT-TEN

Das Programm ist für Extended-Basic geeignet. Um das Programm in TI-Basic laufen zu lassen, brauchen nur die Equates geändert werden. Call Link ("SCROLL", A,B,C[,D,E]) scrollt einen bestimmten Teil des Bildschirms nach unten oder nach oben. Dies ist unter anderem zur Darstellung von Tabellen nützlich; die Kopfzeilen einer Tabelle bleiben so immer erhalten. Die Parameter A,B, C,D,E definieren einen Ausschnitt des Bildschirms. A ist die oberste, B die unterste Zeile, D ist die erste, E die letzte Spalte. D und E sind optional. Falls sie nicht eingegeben werden ist D=1 und E=32. C gibt an, um wieviel Zeilen gescrollt wird. Bei positivem C wird nach unten, bei negativem C nach oben gescrollt.

Heinrich Acker

```
**************************************
*                                    *
*  CALL LINK("SCROLL",A,B,C [,D,E])  *
*                                    *
*                                    *
*  A,B,D,E  DEFINIEREN EINEN BILD-   *
*  SCHIRMAUSSCHNITT,DER C-MAL GE-    *
*  SCROLLT WIRD.                     *
*                                    *
*  Ø<C<25  :NACH UNTEN SCROLLEN      *
*  -25<C<Ø :NACH OBEN SCROLLEN       *
*  A :ERSTE ZEILE                    *
*  B :LETZTE ZEILE                   *
*  C :ANZAHL DES SCROLLENS           *
*  OPTION:                           *
*  D :ERSTE SPALTE     DEFAULT: 1    *
*  E :LETZTE SPALTE    DEFAULT:32    *
*                                    *
*  EXT.BASIC    306 BYTES            *
*                                    *
*  Ø9.Ø7.1985 BY HEINRICH ACKER      *
**************************************
       IDT  'XSCROLLØ'
       DEF  SCROLL
VMBW   EQU  >2Ø24        *
VMBR   EQU  >2Ø2C        *
GPLWS  EQU  >83EØ        *
STATUS EQU  >837C        *
NUMREF EQU  >2ØØC        *
FAC    EQU  >834A        *
XMLLNK EQU  >2Ø18        *
CFI    EQU  >12B8        *
ERR    EQU  >2Ø34        *
ERRSYN EQU  >Ø3ØØ        * SYNTAX ERROR
ERRBV  EQU  >1EØØ        * BAD VALUE
MYWS   BSS  52           *
VDPADR EQU  MYWS+16      * R8
BUFFER EQU  MYWS+18      * AB R9  BUFFER FUER BILDSCHIRMZEILE
SPACE  DATA >8Ø8Ø,>8Ø8Ø,>8Ø8Ø,>8Ø8Ø
       DATA >8Ø8Ø,>8Ø8Ø,>8Ø8Ø,>8Ø8Ø
       DATA >8Ø8Ø,>8Ø8Ø,>8Ø8Ø,>8Ø8Ø
       DATA >8Ø8Ø,>8Ø8Ø,>8Ø8Ø,>8Ø8Ø
********************************
SCROLL LWPI MYWS
       CLR  RØ
       MOVB @>8312,R1    * >8312 ENTHAELT ANZAHL DER PARAMETER
       SRL  R1,8
       LI   R2,14
       CI   R1,5
       JEQ  NUREF        * 5 PARAMETER
       LI   R2,18
       LI   R7,32        * DEFAULT
       LI   R8,1         * DEFAULT
       CI   R1,3
       JEQ  NUREF        * 3 PARAMETER
       LI   RØ,ERRSYN
       BLWP @ERR         * 'SYNTAX ERROR'
NUREF  BLWP @NUMREF      * BRINGT PARAMETER (SPEZ.IN R1) INS FAC
       BLWP @XMLLNK      * WANDELT FAC IN INTEGER UM
       DATA CFI
       MOV  @FAC,@MYWS(R2)
       JEQ  ERBV         * FALLS IRGENDEIN PARAMETER=Ø --> ERBV
       INCT R2
       DEC  R1
       JNE  NUREF        * NAECHSTER PARAMETER
***  R7:E , R8:D , R9:C , R1Ø:B , R11:A  ***************
       C    R8,R7
       JH   ERBV         * FALLS D>E --> ERBV
       CI   R7,32
       JH   ERBV         * FALLS E>32 --> ERBV
       DEC  R8           * R8 :D-1
       C    R11,R1Ø
       JHE  ERBV         * FALLS A>=B --> ERBV
       CI   R1Ø,24
       JH   ERBV         * FALLS B>24 --> ERBV
       DEC  R1Ø          * R1Ø :B-1
       DEC  R11          * R11 :A-1
       LI   R3,32        * R3 :32
       CI   R9,-24
       JHE  CNEG         * FALLS -24<=C<Ø --> CNEG
```

```
        CI    R9,24
        JLE   CPOS          * FALLS Ø<C<=24 --> CPOS
ERBV    LI    RØ,ERRBV      * SONST
        BLWP  @ERR          * 'BAD VALUE'
CNEG    MOV   R11,R4        * R4 :A-1
        MPY   R3,R4         * R5 :(A-1)*32
        NEG   R9            * R9 :ABS(C)
        JMP   LAB1          * --> LAB1
CPOS    MOV   R1Ø,R4        * R4 :B-1
        MPY   R3,R4         * R5 :(B-1)*32
        NEG   R3            * R3 :-32
LAB1    MOV   R1Ø,R4        * R4 :B-1
        S     R11,R4        * R4 :(B-1)-(A-1)=B-A
        MOV   R4,R6
        INC   R6            * R6 :B-A+1
        C     R9,R6
        JH    LAB2          * FALLS ABS(C)>B-A+1 --> LAB2
        MOV   R9,R6         * R6 :ABS(C)
LAB2    A     R8,R5         * R5 :VDP-ADR. 1.ZEILE/1.SPALTE DES DEF.SCHIRMES
        MOV   R7,R2         * R2 :E
        S     R8,R2         * R2 :E-(D-1) = ANZAHL DER ZEICHEN/ZEILE
LOOP1   LI    R1,BUFFER
        MOV   R5,@VDPADR
        MOV   R4,R7
LOOP2   MOV   @VDPADR,RØ    * LOOP2 SCROLLT ZEILENWEISE RUNTER/HOCH.
        A     R3,RØ         * DIES WIRD DURCH LOOP1 ABS(C)-MAL GETAN.
        BLWP  @VMBR
        MOV   @VDPADR,RØ
        BLWP  @VMBW
        A     R3,@VDPADR
        DEC   R7
        JNE   LOOP2
        MOV   @VDPADR,RØ
        LI    R1,SPACE
        BLWP  @VMBW
        DEC   R6
        JNE   LOOP1
        CLR   RØ
        MOVB  RØ,@STATUS
        LWPI  GPLWS
        B     @>ØØ7Ø        * RUECKSPRUNG INS XBAS.
        END
```



LIFE

Das Assembler-Programm ist für das Minimem geschrieben und wird mit der Option RUN unter dem Namen LIFE aufgerufen. Danach wird man gefragt, ob man nach jeder Generation einen Stop haben will. Für die Elementeneingabe dienen die Cursortasten E, S, D und X. Ein Element wird mit der Taste 1 gesetzt und mit der Taste 0 gelöscht. Eine Eingabe beendet man mit Enter. Diese Eingabe bzw. Änderung kann auch während des Ablaufes durch die Quit-Taste aufgerufen werden; die =-Taste dient als Start-/Stoptaste. Das Programm wird mit Enter verlassen. Sterben alle Elemente aus, so stoppt das Programm automatisch.

Frank Rieger

FAST COPY

Dieses Programm läuft unter Editor/Assembler mit 2 Laufwerken. Die Copydiskette muß dabei vorher mit dem Diskmanager initialisiert werden. Das Programm erstellt eine Sektor-Kopie der Masterdiskette, dabei wird ein eventuell vorhandener Inhalt auf der Copydiskette völlig gelöscht. Wer keine zwei Laufwerke besitzt, kann sich das Programm einfach umändern (an den bezeichneten Stellen muß eine entsprechende Meldung mit Tastenabfrage eingefügt werden). Wesentlicher Punkt dieses Programms ist das gezielte Lesen und Schreiben einzelner Sektoren. Um dieses einfacher in andere Programme einbauen zu können, wurde diese Routine als Unterprogramm ausgeführt. Die Wirkungsweise des Programms selber ergibt sich aus den Kommentaren, so daß dazu eigentlich keine weiteren Ausführungen nötig sind, außer der, daß dieses Programm durch eine "Indiskretion" eines Händlers schon länger auf dem „schwarzen" Markt im Umlauf ist und hier nun alle diejenigen in den „Genuß" dessen kommen sollen, die nicht Zugang zu solchen Tauschkanälen haben.

Heiner Martin

```
**********************************
*                                *
*          LIFE                  *
*                                *
**********************************

        DEF  LIFE

VMBW    EQU  >6028
VSBW    EQU  >6024
VSBR    EQU  >602C
KSCAN   EQU  >6020

TMODE   EQU  >8374
ASCII   EQU  >8375
GPLSTA  EQU  >837C

ST      BSS  2
BU      BSS  8
B1      BSS  484
H1      BSS  440

LIFE    JMP  W1
        DATA 100
T0      TEXT 'STOP'
T1      TEXT 'GENERATION: '
T2      TEXT 'ELEMENTE: '
T3      DATA >003C,>4242,>4242,>3C00
        DATA >FFFF,>FFFF,>FFFF,>FFFF
T4      TEXT 'ENTER=ENDE'
T5      TEXT '  =  =STOP'
T6      TEXT 'QUIT=AENDERN'
T7      TEXT 'AUTOMATISCHER STOP ? (J/N)'


W1      MOV  11,12
        LI   0,>0900
        LI   1,T3
        LI   2,>0010
        BLWP @VMBW
        CLR  @ST
        LI   0,B1
        LI   1,>00F2
        CLR  2
M0      MOV  2,*0+
        DEC  1
        JGT  M0
        LI   0,>02FF
        LI   1,>2000
L0      BLWP @VSBW
        DEC  0
        JGT  L0
        JEQ  L0
        LI   0,>0182
        LI   1,T7
        LI   2,>001A
        BLWP @VMBW
A9      CLR  @TMODE
        BLWP @KSCAN
        MOVB @ASCII,0
        SRL  0,8
        CLR  @GPLSTA
        CI   0,>004E
        JEQ  A8
        CI   0,>004A
        JNE  A9
        SETO @ST
A8      LI   0,>0182
        LI   1,>2000
        LI   2,>001A
A7      BLWP @VSBW
        INC  0
        DEC  2
        JGT  A7
        LI   0,>004A
        LI   1,>2200
        LI   2,>0016
C0      BLWP @VSBW
        AI   0,>02A0
        BLWP @VSBW
        AI   0,>FD61
        DEC  2
        JGT  C0
        LI   0,>006A
        LI   2,>0014
C1      BLWP @VSBW
        AI   0,>0015
        BLWP @VSBW
        AI   0,>000B
        DEC  2
        JGT  C1
        LI   0,>0062
        LI   2,>000A
        LI   3,T4
        BL   @PR
        LI   0,>0064
        LI   2,>000A
        LI   3,T5
        BL   @PR
        LI   0,>0086
        LI   2,>000C
        LI   3,T6
        BL   @PR
        BL   @E1
        JMP  W6
E1      LI   0,>006B
        MOV  0,2
        MOV  0,3
        AI   3,>0013
        LI   4,B1
        AI   4,>0017
M8      BLWP @VSBR
        MOV  1,5
        LI   1,>1E00
        BLWP @VSBW
M1      CLR  @TMODE
        BLWP @KSCAN
        MOVB @GPLSTA,6
        SLA  6,3
        JNC  M1
        MOVB @ASCII,6
        SRL  6,8
        MOV  5,1
        BLWP @VSBW
        CLR  @GPLSTA
        CI   6,>0053
        JEQ  M2
        CI   6,>0044
        JEQ  M3
        CI   6,>0045
        JEQ  M4
        CI   6,>0058
        JEQ  M5
        CI   6,>0031
        JEQ  M6
        CI   6,>0030
        JEQ  M7
        CI   6,>000D
        JNE  M8
        B    *11
M2      C    0,2
        JEQ  M8
        DEC  0
        DEC  4
        JMP  M8
M3      C    0,3
        JEQ  M8
        INC  0
        INC  4
        JMP  M8
M4      CI   0,>008B
        JLT  M8
        AI   0,>FFE0
        AI   4,>FFEA
        AI   2,>FFE0
        AI   3,>FFE0
        JMP  M8
M5      CI   0,>02BE
        JGT  M8
        AI   0,>0020
        AI   2,>0020
        AI   3,>0020
        AI   4,>0016
        JMP  M8
M6      LI   7,>0100
        MOVB 7,*4
        LI   1,>2100
        JMP  M9
M7      CLR  7
        MOVB 7,*4
        LI   1,>2000
M9      BLWP @VSBW
        JMP  M8
W6      LI   1,T1
        LI   0,>0002
        LI   2,>000C
        BLWP @VMBW
        LI   1,T2
        LI   0,>0024
        LI   2,>000A
        BLWP @VMBW
        CLR  15
L7      LI   0,B1
        MOV  0,1
        MOV  0,2
        AI   1,>0016
        AI   2,>002C
        LI   4,>0014
        LI   5,H1
        CLR  14
L4      CLR  13
        MOVB 13,*5+
        LI   3,>0014
L3      MOVB *0+,7
        AB   *0+,7
        AB   *0,7
        DEC  0
        AB   *1+,7
        MOVB *1,13
        AB   *1+,7
        AB   *1,7
        DEC  1
        AB   *2+,7
        AB   *2+,7
        AB   *2,7
        DEC  2
        SRL  7,8
        CI   7,>0003
        JEQ  W2
        SRL  13,8
        S    13,7
        CI   7,>0003
        JEQ  W2
        CLR  7
        MOVB 7,*5+
        JMP  W3
W2      LI   7,>0100
        MOVB 7,*5+
        INC  14
W3      DEC  3
        JGT  L3
        INCT 0
        INCT 1
        INCT 2
        CLR  7
        MOVB 7,*5+
        DEC  4
        JGT  L4
        INC  15
        MOV  15,3
        LI   0,>000E
        BL   @ZA
        LI   0,>006B
        MOV  0,2
        LI   3,H1
        INC  3
```

```
        LI   5,>0014
        LI   6,B1
        AI   6,>0017
L6      LI   4,>0014
L5      MOVB *3+,1
        MOVB 1,*6+
        AI   1,>2000
        BLWP @VSBW
        INC  0
        DEC  4
        JGT  L5
        AI   2,>0020
        MOV  2,0
        INCT 3
        INCT 6
        DEC  5
        JGT  L6
        MOV  14,3
        LI   0,>002E
        BL   @ZA
        CI   14,>0000
        JEQ  B0
        MOV  @ST,0
        JNE  B0
        CLR  @TMODE
        BLWP @KSCAN
        MOVB @ASCII,0
        SRL  0,8
        CLR  @GPLSTA
        CI   0,>000D
        JEQ  EN
        CI   0,>0005
        JNE  A1
        BL   @E1
        B    @L7
A1      CI   0,>003D
        JEQ  B8
        B    @L7
B8      LI   0,>7FFF
B9      DEC  0
        JGT  B9
B0      LI   0,>001C
        LI   1,T0
        LI   2,>0004
        BLWP @VMBW
Z9      CLR  @TMODE
        BLWP @KSCAN
        MOVB @ASCII,13
        SRL  13,8
        CLR  @GPLSTA
        CI   13,>000D
        JEQ  EN
        CI   13,>003D
        JEQ  Z0
        CI   13,>0005
        JNE  Z9
Z0      LI   0,>001C
        LI   1,>70BE        * REG 3
        LI   2,>0004
        LI   3,>2020
        MOV  3,4.
        BLWP @VMBW
        CI   13,>0005
        JNE  Z1
        BL   @E1
Z1      B    @L7
EN      B    *12
ZA      LI   4,BU
        LI   5,>2020
        MOV  5,*4+
        MOV  5,*4
        LI   1,BU
        LI   2,>0004
        BLWP @VMBW
        LI   4,>000A
        LI   5,BU
        AI   5,>0007
        LI   6,>3000
        CLR  7
P2      CLR  2
        DIV  4,2
        SLA  3,8
        AB   6,3
        MOVB 3,*5
        INC  7
        DEC  5
        MOV  2,3
        JNE  P2
        MOV  5,1
        INC  1
        MOV  7,2
        BLWP @VMBW
        B    *11
PR      MOVB *3+,1
        BLWP @VSBW
        AI   0,>0020
        DEC  2
        JGT  PR
        B    *11

        END
```

```
**************************************
*                                    *
*             FAST-COPY              *
*                                    *
*  SEKTORKOPIERPROGRAMM FUER 2 DISKS *
*                                    *
*           H.Martin  7/84           *
*                                    *
*                                    *
**************************************


        DEF   START

        REF   VMBW,VMBR,VSBW,VSBR
        REF   VWTR,GPLLNK,DSRLNK
        REF   KSCAN

        AORG >2800          * FUER ASSEMBLER WICHTIG, DA BUFFER
*                             AB >A000
MYWS    BSS  32

START   LWPI MYWS           * EIGENER WORKSPACE
        BL   @CLRSCR        * BILDSCHIRM LOESCHEN
        LI   0,40           * TEXTE AUF DEN BILDSCHIRM
        LI   1,TEXT1
        LI   2,16
        BLWP @VMBW
        LI   0,130
        LI   1,TEXT2
        LI   2,24
        BLWP @VMBW
        LI   0,194
        LI   1,TEXT3
        LI   2,22
        BLWP @VMBW
        LI   0,258
        LI   1,TEXT4
        LI   2,25
        BLWP @VMBW
        LI   0,324
        LI   1,TEXT9
        LI   2,12
        BLWP @VMBW
        BL   @ACTON         * FERTIG, TON
COPY1   LIMI >0002          * WEGEN TON INTERRUPT ZULASSEN
        LIMI >0000
        CLR  0
        MOVB 0,@>837C       * GPL-STATUSBYTE LOESCHEN
        CLR  @>8374         * TASTATURMODUS 0
        BLWP @KSCAN         * TASTATURABFRAGE
        MOVB @>837C,0       * KEINE TASTE GEDRUECKT
        JEQ  COPY1          * DANN VON VORNE
        MOVB @>8375,8
        SRL  8,8
        CI   8,>000D        * ENTER TASTE?
        JEQ  COPY2
        BL   @BADTON        * FALSCHE TASTE, DANN TON
        JMP  COPY1          * VON VORNE
```

```
COPY2  LI   Ø,64          * BILDSCHIRM TEILWEISE LOESCHEN
       CLR  13            * FLAG
       BL   @CLRSC1
       LI   Ø,262         * TEXT 'IN ARBEIT'
       LI   1,TEXT5
       LI   2,9
       BLWP @VMBW
       CLR  4             * SECTOR Ø
       LI   5,>ØCØØ       * VDP-ADRESSE DES BUFFERS
       LI   6,>Ø1ØØ       * NUMMER DES LAUFWERKES
       BL   @RDSEC        * SEKTOR LESEN
       JEQ  ERROR         * BEI FEHLER
       MOVB @>8350,1Ø     * EVENTUELLER FEHLER ABFRAGEN
       JNE  ERROR         * SPRUNG BEI FEHLER
       LI   Ø,>ØCØA
       LI   1,MYWS+14     * ANZAHL SEKTOREN AUS SEKTOR Ø
       LI   2,2           * IN R7 (MYWS+14) LESEN
       BLWP @VMBR
       DEC  7             * MINUS 1, DA SEKTOREN AB Ø
       LI   4,119         * 12Ø SEKTOREN PASSEN INS RAM
       MOV  4,9
COPY8  LI   8,>DØØØ       * BUFFER

* HIER GGF. MELDUNG "MASTERDISK EINLEGEN"
* UND ABFRAGE NACH TASTENDRUCK EINFUEGEN

       LI   6,>Ø1ØØ       * LAUFWERK NUMMER
COPY5  LI   5,>3ØØØ       * BUFFER IM VDP
COPY3  LI   Ø,276         * SEKTORNUMMER ANZEIGEN
       MOV  4,15
       BL   @DISPL
       BL   @RDSEC        * SEKTOR LESEN
       JEQ  ERROR         * ERROR ABFRAGEN
       MOVB @>8350,1Ø
       JNE  ERROR
       DEC  9             * MINUS 1
       DEC  4             * MINUS 1
       AI   5,>FFØØ       * VDP-BUFFER MINUS >1ØØ
       CI   5,>ØEØØ       * VDP VOLL?
       JNE  COPY3         * NEIN, WEITER
       MOV  9,9           * SCHON 12Ø SEKTOREN?
       JLT  COPY4         * JA, SPRUNG
       MOV  8,1           * SKTOREN AUS DEM VDP IN DIE RAM-
       LI   Ø,>ØFØØ       * ERWEITERUNG
       LI   2,>28ØØ
       BLWP @VMBR
       LI   8,>AØØØ       * NEUER POINTER ZUM BUFFER IM RAM
       JMP  COPY5         * DA CAPO

* HIER GGF. MELDUNG "COPYDISK EINLEGEN"
* UND ABFRAGE NACH TASTENDRUCK EINFUEGEN
* DANN WEITER UNTEN LAUFWERKNUMMER 1 EINSETZEN

COPY4  LI   8,>AØØØ       * POINTER ZUM RAM
       INC  4             * NAECHSTER SEKTOR
       INC  9             * DITO
COPY7  LI   5,>ØFØØ       * WIR BEGINNEN RUECKWAERTS
       LI   6,>Ø2ØØ       * LAUFWERKNUMMER 2
COPY6  LI   Ø,276         * SEKTORNUMMER ANZEIGEN
       MOV  4,15
       BL   @DISPL
       BL   @WRSEC        * SCHREIBE SEKTOR
       JEQ  ERROR
       MOVB @>8350,1Ø     * ERRORABFRAGE
       JNE  ERROR
       INC  4             * NAECHSTER SEKTOR
       INC  9             * NAECHSTER SEKTOR
       AI   5,>Ø1ØØ       * VDP-BUFFER NAECHSTER SEKTOR
       CI   5,>37ØØ       * ENDE ERREICHT?
       JNE  COPY6         * NEIN, WEITER
       MOV  8,1           * NAECHSTE SEKTOREN AUS DEM
       LI   Ø,>ØFØØ       * RAM-BUFFER HOLEN
       LI   2,>28ØØ
       BLWP @VMBW
       LI   8,>DØØØ       * POINTER ZUM NAECHSTEN BLOCK
       CI   9,12Ø         * SCHON ALLE?
       JNE  COPY7         * NEIN, WEITER
       DEC  9             * WAREN NUR 12Ø
       AI   4,119         * GESAMTANZAHL
       C    4,7           * REST FESTSTELLEN
       JLT  COPY8         * GROESSER ALS 12Ø, DANN WEITER
       MOV  7,4           * REST DER SEKTOREN
       MOV  13,13         * FLAG PRUEFEN
       JNE  ENDE          * WAR SCHON EINMAL REST, DANN ENDE
       SETO 13            * FLAG SETZEN FUER LETZTEN DURCHLAUF
       JMP  COPY8

ERROR  LI   Ø,196         * TEXT 'ERROR'
       LI   1,TEXT6
       LI   2,11
       BLWP @VMBW
       MOVB 1Ø,1Ø         * SEKTOR-ERROR
       JNE  ERRO1
       MOVB 1,1Ø          * ERRORCODE
ERRO1  AI   Ø,14          * ERRORCODE ANZEIGEN
       MOV  1Ø,1
       SRL  1,12          * HOECHSTWERTIGSTES NYBBLE
       SLA  1,8
       AI   1,>3Ø3Ø       * IN ASCII
       SLA  1Ø,4          * NIEDRIGWERTIGES NYBBLE
       SRL  1Ø,12
       A    1Ø,1
       BLWP @VSBW         * ANZEIGEN
       INC  Ø
       SWPB 1
       BLWP @VSBW         * KOMPLETT
       BL   @BADTON       * TON UND ENDE
       JMP  ENDE1

ENDE   LI   Ø,26Ø         * TEXT ANWEISUNG
       LI   1,TEXT7
       LI   2,23
       BLWP @VMBW
       BL   @ACTON        * TON
ENDE1  LI   Ø,324
       LI   1,TEXT8       * TEXT WEITER
       LI   2,17
       BLWP @VMBW
```

```
ENDE2   LIMI >0002          * INTERRUPT WEGEN TON
        LIMI >0000
        CLR  Ø              * TASTATURABFRAGE WIE OBEN
        MOVB Ø,@>837C
        CLR  @>8374
        BLWP @KSCAN
        MOVB @>837C,Ø
        JEQ  ENDE2
        MOVB @>8375,1
        SRL  1,8
        CI   1,>000C        * TASTE PROCEED?
        JEQ  ENDE3          * JA, WEITER
        BL   @BADTON        * FALSCHE TASTE, TON
        JMP  ENDE2
ENDE3   B    @START


TEXT9   TEXT 'By H. Martin'
        EVEN

*************************
* ACCEPT TONE AUSGEBEN

ACTON   CLR  14
        MOVB 14,@>837C      * GPL-STATUS LOESCHEN
        BLWP @GPLLNK        * TON UEBER GPLLNK AUFRUFEN
        DATA >0034
        RT

*************************
* BAD ACCEPT TON AUSGEBEN

BADTON  CLR  14
        MOVB 14,@>837C      * GPL-STATUS LOESCHEN
        BLWP @GPLLNK        * TON UEBER GPLLNK AUFRUFEN
        DATA >0036
        RT                  * RETURN 11

*************************
* SCREEN LOESCHEN
* HINTERHER:

CLRSCR  CLR  Ø              * SCREEN LOESCHEN
CLRSC1  LI   2,>0300
        LI   1,>2020        * CLRSC1 FUER TEILWEISES LOESCHEN
CLRSC2  BLWP @VSBW          * RØ=ADRESSE, R2=ANZAHL
        INC  Ø              * NAECHSTE BILDSCHIRMADRESSE
        C    Ø,2            * ENDE ERREICHT?
        JNE  CLRSC2         * NEIN, WEITER
        RT                  * RETURN 11

*************************
* READ AND WRITE SECTOR

RDSEC   LI   1,>0100        * KENNWERT FUER READ
        JMP  SEC1
WRSEC   CLR  1              * KENNWERT FUER WRITE
SEC1    MOVB 1,@>834D       * KENNWERT FUER UNTERPROGRAMM SEKTOR
        MOV  4,@>8350       * SEKTORNUMMER
        MOV  5,@>834E       * VDP-ADRESSE FUER BUFFER
        MOVB 6,@>834C       * LAUFWERKNUMMER
        LI   Ø,>0480        * PSEUDO-PAB SCHREIBEN AB >0480 IM VDP
        MOV  Ø,@>8356       * POINTER FUER DSRLNK
        LI   3,>0110        * SUBPROGRAMM NAME >10, LAENGE >01
        LI   2,2
        LI   1,MYWS+6       * R3
        BLWP @VMBW          * "PAB" INS VDP-RAM
        BLWP @DSRLNK        * UND AUFRUFEN
        DATA 10             * FUER SUBPROGRAMME
        B    *11

*************************
* DISPL

DTEN    DATA 10             * FUER DEZIMALWERT

DISPL   CLR  14             * VORBEREITEN FUER DIVISION
        DIV  @DTEN,14       * R14 GETEILT DURCH 10
        AI   15,>0030       * REST IN ASCII VERWANDELN
        SWPB 15
        MOVB 15,1
        BLWP @VSBW          * AUF DEN BILDSCHIRM
        DEC  Ø              * NAECHSTE VDP-ADRESSE (STELLE)
        MOV  14,15          * NOCH ETWAS ZUM TEILEN DA?
        JNE  DISPL          * JA, VON VORNE
        LI   1,>2020        * STELLE DAVOR NOCH LOESCHEN
        BLWP @VSBW
        B    *11

TEXT1   TEXT '**  FASTCOPY  **'
TEXT2   TEXT 'MASTERDISK IN LAUFWERK 1'
TEXT3   TEXT 'COPYDISK IN LAUFWERK 2'
TEXT4   TEXT 'DANN ENTER-TASTE DRUECKEN'
TEXT5   TEXT 'IN ARBEIT'
TEXT6   TEXT 'ERROR-CODE:'
TEXT7   TEXT 'ANWEISUNG ABGESCHLOSSEN'
TEXT8   TEXT 'WEITER MIT PROC'D'

        EVEN

        END
```

MSX®
REVUE
DAS MAGAZIN
FÜR FREUNDE
DER KOMPATIBLEN
DM 5,80/ÖS 49/SFR 5,80

IM TEST:
Spectravideo
Xpress -
wirklich
ein Profi?

IM TEST:
Supersoftware
von Sanyo -
Spiele von
Konami -
Anwenderpro-
gramme von
Microland

SERIE:
Alle MSX-
Basic-Befehle!
Rund 30 Seiten
Listings
für Ihren
MSX-
Computer!

SERVICE
TIPS & TRICKS

GPLLNK

GPLLNK FÜR EXTENDED BASIC

Es hat in der Vergangenheit schon mehrere Vorschläge für ein GPLLNK unter Extended Basic gegeben. Entweder waren diese Programme aber sehr lang, oder sie funktionierten nicht richtig. Sogar die von Texas Instruments USA verschickte Version hatte einen Fehler, man konnte nicht zweimal hintereinander eine der beiden Ton-Routinen aufrufen.
Die hier vorgestellte, sehr kurze Version basiert durchaus auf dem Vorschlag von Texas Instruments, jedoch wurde der Fehler beseitigt und dazu noch das Programm gekürzt. Verwendet wird die Routine genauso wie sonst das GPLLNK im Editor/Assembler-Modul bzw. Mini-Memory, d.h. also nach dem BLWP-Aufruf kommt ein folgendes DATA mit der Grom-Adresse der aufzurufenden Routine. Beim Assemblieren wird es einfach an das Programm, welches diese Routine benötigt, hinten angehängt. Anmerken muß ich hier noch, daß dieses GPLLNK ausschließlich unter Extended Basic funktioniert.

Heiner Martin

```
*****************************************
*                                       *
*     GPLLNK fuer EXTENDED BASIC        *
*                                       *
*          3.7.84 H. Martin             *
*                                       *
*****************************************

*       EQUATES

UTLWS   EQU   >2038          * UTILITIE WORKSPACE
SUBST   EQU   >8373          * SUBSTACK POINTER
GRMRA   EQU   >9802          * GROM READ ADRESSE
GPWS    EQU   >83E0          * GPL-WORKSPACE

*       EINSPRUNGVEKTOREN

GPLLNK  DATA  UTLWS
        DATA  GPLLN1

*       PROGRAMM

GPLLN1  MOVB  @GRMRA,0       * GROM ADRESSE LESEN
        SWPB  0
        MOVB  @GRMRA,0
        SWPB  0
        AI    0,-3           * ZURUECK ZUM XML BEFEHL VON CALL LINK
        MOVB  @SUBST,1       * GPL-SUBSTACK POINTER
        SRL   1,8
        AI    1,>8300        * KOMPLETTE ADRESSE DES SUBSTACK
        INCT  1              * XML ADRESSE FUER RETURN AUF
        MOV   0,*1           * DEN STACK LADEN
        SWPB  1
        MOVB  1,@SUBST       * NEUER SUBSTACK-POINTER
        LI    3,>2000
        MOV   *3,2           * RETTE XML LINK
        LI    0,GPLLN2       * NEUES XML LINK
        MOV   0,*3           * AUF DEN XML-POINTER
        MOV   *14+,@>83EC    * GPL-ADRESSE (DATA) IN R6 GPLWS
        LWPI  GPWS           * GPL-WORKSPACE LADEN
        B     @>0060         * ROUTINE AUFRUFEN MIT RESET GPL
GPLLN2  LWPI  UTLWS
        MOV   2,*3           * ALTES XML LINK WIEDERHERSTELLEN
        RTWP                 * UND ZURUECK
```

BIT MAP MODE

Das vorliegende Bit-Map-Mode-Programm erlaubt es Ihnen, mittels 16 Befehlen Grafiken im Bit-Map-Mode, dem höchsten Auflösungs-Mode auf dem TI 99/4A, zu erzeugen, abzuspeichern, zu laden, auszudrucken und sich auf dem Bildschirm anzuschauen.

Bevor ich auf die einzelnen Befehle eingehe, möchte ich noch ein paar Anmerkungen zu meinem Programm loslassen.

Die Speichererweiterung ist trotz Minimem's Voraussetzung. Die Grafiken werden blind erzeugt, d.h. die Grafik wird in einem speziellen Speicherbereich in der Speichererweiterung erstellt und kann mittels eines Befehles für eine bestimmte Zeit auf dem Bildschirm gezeigt werden. Danach kehrt das Maschinenprogramm zum Titelbildschirm zurück. Das Basic-Programm zur Erzeugung der Grafik ist somit gelöscht, dies ist auch schon der Nachteil des Bit-Map-Mode's. Sie können sich die Grafik allerdings immer wieder von Basic her anschauen und verändern, da diese in der Speichererweiterung erhalten bleibt. Ein Diskettenlaufwerk ist wünschenswert, da das Programm die Abspeicherung der Grafiken nur auf Diskette unterstützt. Auch eine Hardcopy für den Seikosha GP 100 A ist vorhanden.
Alle Befehle werden mittels CALL LINK und eines Programmnamens aufgerufen. Zum Teil müssen Parameter mit übergeben werden. Es sind nur Strings, Stringvariablen, Zahlen und numerische Variablen erlaubt.
Falsche Parameter führen zu einer Fehlermeldung und Programmabbruch.

Befehlssatz:

CALL LINK ("CLEAR") —>
Dieser Befehl löscht die Grafik. Er sollte als erstes in Ihrem Programm stehen.

CALL LINK ("SETPIX", Y,X,(bis zu 7 mal))—>
Dieser Befehl setzt an die Koordinate X,Y einen Punkt. Es können bis zu sieben Punkte gleichzeitig gesetzt werden.

CALL LINK ("CLRPIX", Y,X,(bis zu 7 mal))—>
Dieser Befehl löscht einen Punkt mit der Koordinate X,Y. Es können bis zu 7 Punkte gleichzeitig gelöscht werden.

CALL LINK ("WRITE", A, Zeile, Spalte, STRING)—>
Dieser Befehl ermöglicht es, Texte in die Grafik zu schreiben. Wenn A gleich 1 ist, dann kann man den Text in die Grafik hineinkopieren und wenn A gleich 0 ist, dann wird die Grafik unter dem Text gelöscht. Die Textposition wird durch Zeile und Spalte bestimmt. Es sind Strings mit einer Länge von bis zu 32 Zeichen erlaubt. Zur Verfügung ste-

Bitte weiter auf S. 61

```
      UNL
* ** BIT-MAP-GRAFIC
* *** ================
* **** (c) by Bernd Bertling, Neu-Crengeldanzstra~e 2
* ***** 4600 Dortmund 72
* ******
* ***** Programmidentifizierung
* ****    IDT  'BITMGRFC'
* ***** Bei Erstellung eines Objektcodes f)r Editor/Assembler m)~en folgende
* ***** Refferenzen statt der unten genannten Equates geschrieben werden.
* ***** Refferenzen f)r Editor/Assembler
* ******      REF  VSBW,VSBR,VMBW,VMBR,VWTR,ERR
* ******      REF  XMLLNK,NUMREF,STRREF,DSRLNK,GPLLNK
* ******      AORG >A000  Absolute-Code f)r Minimem
* ***** Programmnamen Definition
* ****      DEF  CLEAR,SETPIX,CLRPIX,ZEIG,WRITE,CHAR,COLOR,SAVE
* ***       DEF  OLD,HCOPY,KREIS,CKREIS,LINE,CLINE,WLINE,CWLINE
* ** Equates f)r Minimem
* *
VSBR   EQU  >602C      Ein Byte aus dem VDP-RAM lesen
VSBW   EQU  >6024      Ein Byte in's VDP-RAM schreiben
VMBR   EQU  >6030      Mehrere Bytes aus dem VDP-RAM lesen
VMBW   EQU  >6028      Mehrere Bytes in's VDP-RAM schreiben
VWTR   EQU  >6034      VDP-Register ver(ndern
XMLLNK EQU  >601C      Verbindung zu ROM-Routinen
NUMREF EQU  >6044      Numerische Parameter: Basic --> Assembler
STRREF EQU  >604C      Strings: Basic --> Assembler
ERR    EQU  >6050      Fehlermeldung: Assembler --> Basic
DSRLNK EQU  >6038      Verbindung zur DSR-Routine
GPLLNK EQU  >6018      Verbindung zu GROM-Routinen
* ** Sonstige Equates
* *
SIN    EQU  >002E      GPL-Routine --> Sinus-Funktion
COS    EQU  >002C      GPL-Routine --> Cosinus-Funktion
RES    EQU  >0038      Platz im VDP-RAM reservieren
FADD   EQU  >0600      ROM-Routine --> Gleitkomma Addition
FMULT  EQU  >0800      ROM-Routine --> Gleitkomma Multiplikation
FDIV   EQU  >0900      ROM-Routine --> Gleitkomma Division
CFI    EQU  >1200      Gleitkomma --> Integer
* Folgender Wert mu~ bei der Editor/Assembler-Version von >7200 in
* >2300 ge(ndert werden.
CIF    EQU  >7200      Integer --> Gleitkomma
ERR22  EQU  >1600      'BAD-ARGUMENT' - Fehlermeldung
LNKNUM EQU  >8312      Anzahl der Variablen in CALL LINK
WATCH  EQU  >8379      VDP-Interrupt-Timer
GPLSTA EQU  >837C      Statusregister
FAC    EQU  >834A      Gleitkomma Accu
ARG    EQU  >835C      Gleitkomma Argument
```

```
VCOP1  EQU  >83D4      Kopieadresse des VDP-Registers 1
GPLWS  EQU  >83E0      GPL-Arbeitsbereich
POINT  EQU  >831C      Pab-Pointer
PNTR   EQU  >8356      DSR-Namenszeiger
*
* Variablen
*
PABBUF BSS  2    Variable f}r Pab-Buffer-Adresse
PAB    BSS  2    Variable f}r Pab-Adresse
PAB5   BSS  2    Adressen-Variable
BACK1  BSS  2    Returnadressen-Buffer f}r Unterprogramme
BACK2  BSS  2    Returnadressen-Buffer f}r Unterprogramme
MYREG1 BSS  32   Eigene Arbeitsregister
MYREG2 BSS  32   2.Satz Arbeitsregister
HELP1  BSS  2    Hilfs-Variablen
HELP2  BSS  2      "  -    "
HELP3  BSS  2      "  -    "
STRIN0 BSS  8      "  -    "
STRIN1 BSS  8      "  -    "
STRIN2 BSS  8      "  -    "
SATZ   BSS  33   33 Byte Variablen-Buffer
PRMZAL BSS  2    Schleifenz{hler
ANZAHL BSS  2        "
MATRIX BSS  56   Hilfsvariablen-Feld
P      BSS  2    Pixel setzen oder l|schen
BUFADR BSS  2    Buffer f}r aktuelle Buffer-Adresse
PATT   BSS  6144 Pattern-Description-Buffer
FARBE  BSS  6144 Farblisten-Buffer
BUFFER BSS  768  Standard-ASCII-Buffer
FPBUF1 BSS  8
FPBUF2 BSS  8
XAFP   BSS  8
YAFP   BSS  8
XEFP   BSS  8
YEFP   BSS  8
WINKEL BSS  8    Beinhaltet den aktuellen Winkel in Radiant
*
XA     BSS  2
YA     BSS  2
XE     BSS  2
YE     BSS  2
X      BSS  2    Hilfsvariable
Y      BSS  2        "
RQ     BSS  2        "
*
* Datei-Daten f}r Disketten-Datei
*
PDATA  DATA >0000,>0000,>5000,>0000,>0000
*
* Datei-Daten f}r Drucker-Datei
*
PDATA2 DATA >0012,>0000,>8000,>0000,>0006
       TEXT 'PIO.CR'
MYDAT1 BYTE >0A,>1B,>10,>00,>68
MYDAT2 BYTE >1B,>10,>00,>E8
MYDAT3 BYTE >0A,>0A,>0A,>0A,>0A
*
* VDP-Register
*
VREG1  DATA >0002,>01E0,>0206,>03FF,>0403,>0536,>0600,>0712  Bitmap-Mode
VREG2  DATA >0000,>01E0,>0200,>030C,>0400,>0506,>0600,>0707  Grafic-Mode
*
* Konstanten
*
BYTM55 BYTE -55
BYTM48 BYTE -48
BYT0   BYTE 0
BYT1   BYTE 1
BYT2   BYTE 2
WRITEE BYTE 3
BYT15  BYTE 15
BYT16  BYTE 16
BYT32  BYTE 32
BYT50  BYTE 50
BYT128 BYTE 128
       EVEN
D01    DATA 1
D02    DATA 2
D05    DATA 5
D08    DATA 8
D09    DATA 9
D16    DATA 16
D25    DATA 25
D32    DATA 32
D80    DATA 80
D105   DATA 105
D128   DATA 128
D152   DATA 152
D192   DATA 192
D256   DATA 256
D16257 DATA 16257
DIVIS0 DATA >4039,>1D39,>5000,>0000    gleich 57.29578 (Gleitkommazahl)
*
* Weitere Drucker-Daten
*
ZLDATA DATA 0,0,STRIN2,>0007
       DATA 0,256,STRIN1+7,>0106
       DATA 256,512,STRIN1+6,>0205
       DATA 512,768,STRIN1+5,>0304
       DATA 768,1024,STRIN1+4,>0403
       DATA 1024,1280,STRIN1+3,>0502
       DATA 1280,1536,STRIN1+2,>0601
       DATA 1536,1536,STRIN1+1,>0700
       DATA 1792,1792,STRIN2,>0007
       DATA 1792,2048,STRIN1+7,>0106
       DATA 2048,2304,STRIN1+6,>0205
       DATA 2304,2560,STRIN1+5,>0304
       DATA 2560,2816,STRIN1+4,>0403
       DATA 2816,3072,STRIN1+3,>0502
       DATA 3072,3328,STRIN1+2,>0601
       DATA 3328,3328,STRIN1+1,>0700
       DATA 3584,3584,STRIN2,>0007
       DATA 3584,3840,STRIN1+7,>0106
       DATA 3840,4096,STRIN1+6,>0205
       DATA 4096,4352,STRIN1+5,>0304
       DATA 4352,4608,STRIN1+4,>0403
       DATA 4608,4864,STRIN1+3,>0502
       DATA 4864,5120,STRIN1+2,>0601
```

```
       DATA 5120,5120,STRIN1+1,>0700
       DATA 5376,5376,STRIN2,>0007
       DATA 5376,5632,STRIN1+7,>0106
       DATA 5632,5888,STRIN1+6,>0205
       DATA 5888,5888,STRIN1+5,>0300
*
* Wurzel-Routinen-Daten
*
SQRDA1 DATA 0,0,2,1,6,2,12,3,20,4,30,5
       DATA 42,6,56,7,72,8,90,9,110,10
       DATA 132,11,156,12,182,13,210,14
       DATA 240,15,272,16,306,17,342,18
       DATA 380,19,420,20,462,21,506,22
       DATA 552,23,600,24,650,25,702,26
       DATA 756,27,812,28,870,29,930,30
       DATA 992,31,1056,32,1122,33,1190,34
       DATA 1260,35,1332,36,1406,37
       DATA 1482,38,1560,39,1640,40
       DATA 1722,41,1806,42,1892,43
       DATA 1980,44,2070,45,2162,46
       DATA 2256,47,2352,48,2450,49
       DATA 2550,50,2652,51,2756,52
       DATA 2862,53,2970,54,3080,55
       DATA 3192,56,3306,57,3422,58
       DATA 3540,59,3660,60,3782,61
       DATA 3906,62,4032,63,4160,64
       DATA 4290,65,4422,66,4556,67
       DATA 4692,68,4830,69,4970,70
       DATA 5112,71,5256,72,5402,73
       DATA 5550,74,5700,75,5852,76
       DATA 6006,77,6162,78,6320,79
       DATA 6480,80,6642,81,6806,82
       DATA 6972,83,7140,84,7310,85
       DATA 7482,86,7656,87,7832,88
       DATA 8010,89,8190,90,8372,91
       DATA 8556,92,8742,93,8930,94
       DATA 9120,95,9312,96,9506,97
       DATA 9702,98,9900,99,10100,100
       DATA 10302,101,10506,102,10712,103
       DATA 10920,104,11130,105,11342,106
       DATA 11556,107,11772,108,11990,109
       DATA 12210,110,12432,111,12656,112
       DATA 12882,113,13110,114,13340,115
       DATA 13572,116,13806,117,14042,118
       DATA 14280,119,14520,120,14762,121
       DATA 15006,122,15252,123,15500,124
       DATA 15750,125,16002,126,16256,127
SQRDA2 DATA 16512,128,16770,129,17030,130
       DATA 17292,131,17556,132,17822,133
       DATA 18090,134,18360,135,18632,136
       DATA 18906,137,19182,138,19460,139
       DATA 19740,140,20022,141,20306,142
       DATA 20592,143,20880,144,21170,145
       DATA 21462,146,21756,147,22052,148
       DATA 22350,149,22650,150,22952,151
       DATA 23256,152,23562,153,23870,154
       DATA 24180,155,24492,156,24806,157
       DATA 25122,158,25440,159,25760,160
       DATA 26082,161,26406,162,26732,163
       DATA 27060,164,27390,165,27722,166
       DATA 28056,167,28392,168,28730,169
       DATA 29070,170,29412,171,29756,172
       DATA 30102,173,30450,174,30800,175
       DATA 31152,176,31506,177,31862,178
       DATA 32220,179,32580,180,32942,181
       DATA 33306,182,33672,183,34040,184
       DATA 34410,185,34782,186,35156,187
       DATA 35532,188,35910,189,36290,190
       DATA 36672,191,37056,192,37442,193
       DATA 37830,194,38220,195,38612,196
       DATA 39006,197,39402,198,39800,199
       DATA 40200,200,40602,201,41006,202
       DATA 41412,203,41820,204,42230,205
       DATA 42642,206,43056,207,43472,208
       DATA 43890,209,44310,210,44732,211
       DATA 45156,212,45582,213,46010,214
       DATA 46440,215,46872,216,47306,217
       DATA 47742,218,48180,219,48620,220
       DATA 49062,221,49506,222,49952,223
       DATA 50400,224,50850,225,51302,226
       DATA 51756,227,52212,228,52670,229
       DATA 53130,230,53592,231,54056,232
       DATA 54522,233,54990,234,55460,235
       DATA 55932,236,56406,237,56882,238
       DATA 57360,239,57840,240,58322,241
       DATA 58806,242,59292,243,59780,244
       DATA 60270,245,60762,246,61256,247
       DATA 61752,248,62250,249,62750,250
       DATA 63252,251,63756,252,64262,253
       DATA 64770,254,65280,255,65535,256
*
* Copy Directives
*
       COPY "DSK1.@BITMAP1S"    * Unterprogramme 1.Teil
       COPY "DSK1.@BITMAP2S"    * Unterprogramme 2.Teil
       COPY "DSK1.@BITMAP3S"    * Hauptprogramme 1.Teil
       COPY "DSK1.@BITMAP4S"    * Hauptprogramme 2.Teil
*
* Ende des Programms
*
       END  CLEAR   Automatische Buffer-Lischung
```

```
* Segment "@BITMAP1S"
*
* ******************
* *                *
* * Unterprogramme *
* *                *
* *     1.Teil     *
* *                *
* ******************
*
* BL @PIXEL
*
* Diese Routine setzt an einen, durch Koordinaten definierten Punkt, Pixel.
* Die }bergebenen Werte sind:Y-Koordinate ; X-Koordinate . Setzen oder l!schen
* eines Pixels wird durch den Wert in 'P' bestimmt.Steht eine 1 in 'P', so
* wird ein Punkt gesetzt und bei Ø gel!scht.
*
PIXEL  CLR  R1          R1=Ø
       MOVB @ANZAHL,R1  Anzahl der Parameter in CALL LINK nach R1
       SWPB R1          Byte in Wort verwandeln
       MOV  R1,@ANZAHL  Anzahl der Parameter als Wort zur}ck
       MOV  R1,@PRMZAL  Wert kopieren
*
       CI   R1,2        Sind mind. zwei Parameter vorhanden ?
       JLT  ERRT1       Wenn weniger, Fehlermeldung
       CI   R1,14       Sind max. vierzehn Parameter vorhanden ?
       JGT  ERRT1       Wenn mehr, Fehlermeldung
       JMP  PIXEL1      Wenn alles OK, Koordinaten }bernehmen
*
ERRT1  LIMI 2           VDP-Interrupt wieder zulassen
       B    @BARERR     Fehlermeldung ausgeben
*
PIXEL1 LI   R1,1        R1=1
       LI   R6,SATZ     R6 mit Parameter-Buffer-Adresse laden
*
PIXEL2 BL   @TSTGR      Parameter auf Zul{~igkeit pr}fen (Zeile)
       DATA 1,192       Untere Grenze, Obere Grenze
       MOV  @FAC,*R6+   Parameter in Buffer bringen und Adresse erh!hen
       INC  R1          Parameternummer um eins erh!hen
*
       BL   @TSTGR      Parameter auf Zul{~igkeit pr}fen (Spalte)
       DATA 1,256       UG, OG
       MOV  @FAC,*R6+   Parameter in Buffer bringen und Adresse erh!hen
       INC  R1          Parameternummer um eins erh!hen
*
       DECT @ANZAHL     Alle Parameter geholt ?
       JNE  PIXEL2      Wenn nicht, die n{chsten zwei Parameter holen
*
SETTO  MOV  R11,@BACK1
*
       LI   R15,SATZ    R15 mit Parameter-Buffer-Adresse laden
*
PIXEL3 MOV  *R15+,R2    Y-Koordinate }bernehmen (Wort)
       MOV  *R15+,R4    X-Koordinate }bernehmen (Wort)
*
       CB   @BYT1,R7
       JNE  PIXEL4
       C    R2,@DØ1     Y-Koordinate, Untere Grenze
       JLT  PIXELA
       C    R4,@DØ1     X-Koordinate, Untere Grenze
       JLT  PIXELA
       C    R2,@D192    Y-Koordinate, Obere Grenze
       JGT  PIXELA
       C    R4,@D256    X-Koordinate, Obere Grenze
       JGT  PIXELA
*
PIXEL4 DEC  R2          Y-Koordinate=R2-1
       DEC  R4          X-Koordinate=R4-1
       MOV  R2,R6       Y-Koordinate kopieren
       MOV  R4,R14      X-Koordinate kopieren
       SRL  R6,3        R6=R6/8
       MOV  R6,R12      R12=R6
       SLA  R6,3        R6=R6*8
       S    R6,R2       R2=R2-R6
       SLA  R12,8       R12=R12*256
       A    R12,R2      R2=R2+R12
       MOV  R2,RØ       In RØ steht jetzt einen Teil der fertigen Adresse
       SRL  R4,3        R4=R4/8
       SLA  R4,3        R4=R4*8
       S    R4,R14      R14=R14-R4
       A    R4,RØ       In RØ steht die fertige P.D.T.-Buffer-Adresse
       LI   R11,PATT    P.D.T.-Buffer-Adresse in R11 laden
       A    R11,RØ      Basisadresse zum berechneten Wert addieren
       MOV  RØ,R1Ø      Fertige Adresse nach R1Ø kopieren
       CLR  R12         R12=Ø
       MOVB *R1Ø,R12    Byte nach R1 bringen
*
       MOV  R14,RØ      Bitadresse nach RØ, da RØ gleichzeitig Schiebewert ist
       SWPB R12         Byte in Wort verwandeln
       MOV  R12,R9      Grafic-Wert kopieren
       MOV  RØ,RØ       Ist RØ=Ø ?
       JEQ  PIXEL5      Wenn ja, dann braucht nicht geschoben zu werden
       SLA  R9,Ø        R9 um Wert von RØ nach links schieben (Multiplizieren)
PIXEL5 SLA  R9,8         Wert in den sign. Bereich schieben
       SRL  R9,7         Wert um 7 Bits nach rechts (Dividieren)
       CB   @BYT1,R9     Ist das Pixel bereits gesetzt ?
       JNE  PIXEL7       Wenn nicht, dann zu NXT1 springen
*
       CB   @BYT1,@P     Ist das Pixel bereits gesetzt ?
       JEQ  PIXELA       Wenn ja, dann braucht man es nicht setzen
*
       LI   R13,128      R13=128
       MOV  RØ,RØ        Ist RØ=Ø ?
       JEQ  PIXEL6       Wenn ja, dann braucht nicht geschoben zu werden
       SRL  R13,Ø        Sonst um Wert von RØ nach rechts schieben (Dividieren)
PIXEL6 S    R13,R12      R12=R12-R13
       JMP  PIXEL9       N{chste Runde
*
PIXEL7 CB   @BYTØ,@P     Ist das Pixel }berhaupt gesetzt ?
       JEQ  PIXELA       Wenn nicht, dann braucht man es nicht l!schen
*
       LI   R13,128      R13=128
       MOV  RØ,RØ        Ist RØ=Ø ?
       JEQ  PIXEL8       Wenn ja, dann braucht nicht geschoben zu werden
       SRL  R13,Ø        Sonst um Wert von RØ nach rechts schieben (Dividieren)
PIXEL8 A    R13,R12      R12=R12+R13
*
PIXEL9 SWPB R12          Wort in Byte verwandeln
```

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE

TI REVUE

Das Magazin für TI 99-4A

KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre TI REVUE nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Ihr" Kiosk nicht beliefert wurde? Kein Problem! Für ganze 60 DM liefern wir per Post zwölf Hefte ins Haus (Ausland 80 DM). Einfach den Bestellschein auf der nächsten Seite ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). TI REVUE kommt dann pünktlich ins Haus.

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen– Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für zwölf Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

DAS ANGEBOT: KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen ab sofort die TI-REVUE: KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!

Wir veröffentlichen nur Kleinanzeigen privater Inserenten, keine gewerblichen Anzeigen. Die kosten pro Millimeter DM 5,– plus Mehrwertsteuer!

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!

Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!

Die Insertion ist nicht vom Kauf des Heftes abhängig!

RESERVIERUNGS-SERVICE

Selbstverständlich denken wir bei diesem Assembler-Heft auch wieder an jene unter unseren Lesern, die keine Zeit haben, diese ganzen Listings einzugeben. Für sie hält der Kassettenservice das gesamte Angebot dieses Heftes auf drei Disketten bereit. Diese Disketten sind nur im Paket zu beziehen, sie kosten geschlossen 75,– DM. Einfach den Coupon auf der nächsten Seite ausschneiden, ausfüllen und absenden. Diese Seite ist so gestaltet, daß keine Information des Heftes verloren geht.

Ein Versand auf Kassette ist, wie Ihnen sicher bekannt, bei Assemblerprogrammen nicht möglich. Achtung: Volles Umtauschrecht bei Diskettenfehlern! Wir weisen ausdrücklich darauf hin, daß die Disketten nur nach Bestelleingang von Hand gefertigt werden und nicht in jedem Fall vorrätig sind. Deswegen kann es zu Lieferzeiten von bis zu zwei Wochen kommen.

ABO SERVICE-KARTE

TI

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

TI REVUE

Abo-Service
Postfach 1107
8044 UNTERSCHLEISSHEIM

Coupon

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten zwölf Ausgaben an untenstehende Anschrift. Sollte ich nicht vier Wochen vor Ablauf schriftlich kündigen, läuft diese Abmachung automatisch weiter.

Name _____

Vorname _____

Straße/Hausnr. _____

Plz/Ort _____

Ich bezahle:

☐ per beiliegendem Verrechnungsscheck

☐ gegen Rechnung

☐ bargeldlos per Bankeinzug von meinem Konto

bei (Bank) und Ort _____

Kontonummer _____

Bankleitzahl _____

(steht auf jedem Kontoauszug)

Unterschrift _____

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift _____

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen

die Listings dieses Heftes auf

☐ 3 Disketten zum Preis von DM 75,–

Ich zahle:
per beigefügtem Scheck ( )
Gegen Bankabbuchung am Versandtag ( )
Zutreffendes bitte ankreuzen!

Meine Bank (mit Ortsname) .....

Meine Kontonummer .....

Meine Bankleitzahl ..... **(steht auf jedem Bankauszug)**

Vorname ..... **Nachname** .....

Str./Nr. ..... **PLZ / Ort** .....

Hiermit bestätige ich mit meiner Unterschrift, Ihre Verkaufsbedingungen gelesen zu haben und zu akzeptieren.

Unterschrift .....

Bitte ausschneiden und einsenden an
TI-REVUE
Assembler-Service
Postfach 1107
8044 Unterschleißheim

TI REVUE

Das Magazin für TI 99-4A

30 Seiten Listings für Ihren TI 99/4A

Assembler leicht gemacht

Neue Software im Test

Drucker - richtig behandelt!

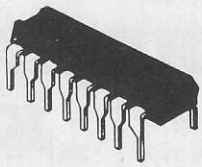
4 Seiten Anzeigen rund um den TI 99/4A

Es geht! Dateien eröffnen und bearbeiten mit dem Kassetten-Recorder

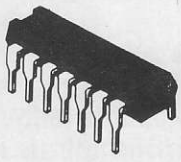
TI-REVUE jeden Monat neu

Gehäuse und thermische Angaben

PLASTIK

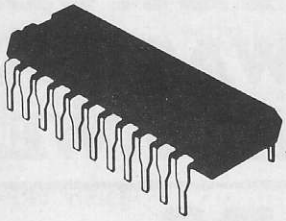


16 PIN - N

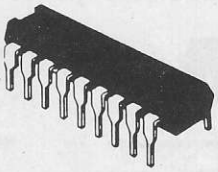


14 PIN - N

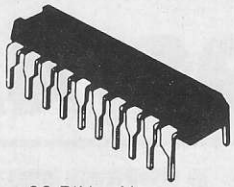
8 PIN - N



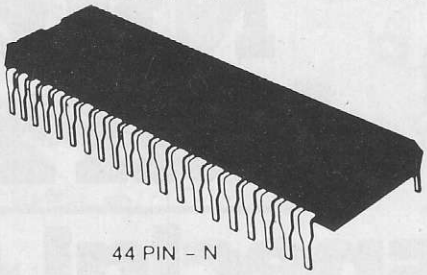
22 PIN - N



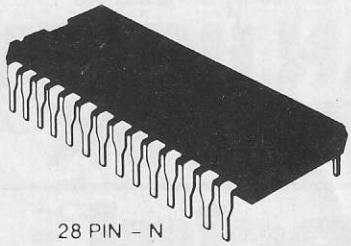
18 PIN - N



20 PIN - N



44 PIN - N



28 PIN - N

ll + special + special + special

TI 99/4A SPECIAL

SONDERHEFT NR. 2/85
DM 14,80/ÖS 124/SFR 14,80

Anwender-Programme
Utilities
Schul-Programme
Spiele
Adventures

SPIEL-SALON
FÜR BUCHHALTER
MUSIKER &
SAMMLER
MATHE-TABELLEN
TORE
DIE HANDWERKER
KOMMEN
FÜR JEDEN ETWAS
WAS MAN SCHWARZ
AUF WEISS BESITZT

Rund 150 Seiten Listings für den 99/4A

JETZT AN IHREM KIOSK

Nur noch kurze Zeit!

special + special + special + s

```
       MOVB R12,*R1Ø    Byte in den Pattbuffer zur}ckschieben
*
PIXELA DECT @PRMZAL     Alle }bergebenen Werte berechnet ?
       JNE  PIXEL3      Wenn nicht, n{chstes Pixel
*
       CB   @BYT1,R7
       JNE  PIXELB
       MOV  @BACK1,R11
       RT
*
PIXELB LIMI 2           VDP-Interrupt wieder zulassen
       B    @BASIC      N{chste Basic Anweisung
*
* BL @BITMAP
*
* Videoprozessor auf BIT-MAP-MODE umschalten, den Pattbuffer in's VDP-
* -RAM bringen und Farbliste setzen.
*
BITMAP MOV  R11,@BACK1  Returnadresse sichern
*
       LI   RØ,>2ØØØ    Farbliste auf gleiche V- und H-Farbe setzen,
       LI   R1,>22ØØ    damit beim {ndern der VDP-Register nichts zu
       LI   R2,6144     sehen ist.
BITMP1 BLWP @VSBW       Byte in's VDP-RAM bringen
       INC  RØ          Adresse plus eins
       DEC  R2          Alle Bytes gesetzt ?
       JNE  BITMP1      Wenn nicht, n{chstes Byte
*
       MOVB @VREG1+3,@VCOP1  VDP-Register 1 kopieren
       LI   R1,VREG1         Adresse der BIT-MAP-MODE Werte
       LI   R2,8        Acht Byte sind zu {ndern
BITMP2 MOV  *R1+,RØ     Registernummer- und Wert nach RØ
       BLWP @VWTR       Register {ndern
       DEC  R2          Alle Register ge{ndert ?
       JNE  BITMP2      Wenn nicht, n{chstes Register
*
* Bitmap-Namensliste setzen
*
       LI   RØ,>18ØØ    Adresse der Namensliste laden
       LI   R1,Ø        R1 = Zeichen Ø
       LI   R2,3        Drei Drittel setzen
BITMP3 BLWP @VSBW       Zeichen }bertragen
       INC  RØ          Adresse um eins erh!hen
       SWPB R1          Byte in Wort verwandeln
       INC  R1          Zeichen um eins erh!hen
       CI   R1,>Ø1ØØ    Ist R1 = 256 ?
       JEQ  BITMP4      Wenn ja, pr}fen ob noch ein Drittel frei ist
*
       SWPB R1          Wort in Byte verwandeln
       JMP  BITMP3      N{chstes Zeichen
BITMP4 LI   R1,Ø        Zeichencode wieder auf null zur}cksetzen
       DEC  R2          Alle Drittel gesetzt ?
       JNE  BITMP3      Wenn nicht, n{chstes Drittel
*
       CLR  RØ          Grafic ab Adresse >ØØØØ in's VDP-RAM bringen
       LI   R1,PATT     Adresse des P.D.T.-Buffers
       LI   R2,6144     6144 (>18ØØ) Bytes sind zu }bertragen
       BLWP @VMBW       Schreiben
*
       LI   RØ,>2ØØØ    Farbliste ab Adresse 8192 in's VDP-RAM bringen
       LI   R1,FARBE    Adresse des Farblisten-Buffers
       LI   R2,6144     6144 (>18ØØ) Bytes sind zu }bertragen
       BLWP @VMBW       Schreiben
*
       MOV  @BACK1,R11  Returnadresse wiedergewinnen
       RT               Return zum rufenden Programm
*
* BL @ENDE
*
* Diese Routine setzt die alten VDP-Register und die alten Werte in's
* VDP-RAM.
*
ENDE   MOVB @VREG2+3,@VCOP1  VDP-Register 1 kopieren
       LI   R1,VREG2         R1 mit Adresse der Registerwerte f}r G.-Mode laden
       LI   R2,8             Acht Register m}~en ge{ndert werden
*
ENDE1  MOV  *R1+,RØ     Registernummer und Wert nach RØ
       BLWP @VWTR       Register {ndern
       DEC  R2          Alle Register ge{ndert ?
       JNE  ENDE1       Wenn nicht, n{chstes Register
*
       LI   RØ,1Ø24      Standard-Zeichensatz wieder im VDP-RAM plazieren
       LI   R1,BUFFER   In R1 steht die Adresse des Buffer's
       LI   R2,768      Der Buffer ist 768 Bytes lang
       BLWP @VMBW       Bytes in's VDP-RAM schreiben
*
       MOVB @BYTØ,@>83C2 Wirkung der QUIT-Taste wiederherstellen
       LIMI 2            VDP-Interrupt Wieder zulassen
       LWPI GPLWS        GPL-Arbeitsregister laden
       BLWP @>ØØØØ       Zum Titelbildschirm zur}ckspringen
*
* BL @STRING
*
* Diese Routine }bernimmt Zeile und Spalte und setzt an diese
* Position im P.D.T.- oder Farb-Buffer einen Acht-Byte-String der aus
* den 16 Bytes gebildet wird, welche von Basic her }bergeben wurden.
*
STRING MOVB @LNKNUM,R12 Anzahl der Parameter nach R12
       LI   R1,1        1.Parameter
       BL   @TSTGR      Parameter }bernehmen und testen
       DATA 1,24
       MOV  @FAC,R13    Zeile
*
       INC  R1          2.Parameter
       BL   @TSTGR      Parameter }bernehmen und testen
       DATA 1,32
       MOV  @FAC,R14    Spalte
*
       CLR  RØ          RØ=Ø
       INC  R1          3.Parameter
       LI   R2,SATZ     R2 mit Stringbuffer-Adresse laden
       MOVB @BYT16,*R2  Maximal 16 Bytes }bernehmen
       BLWP @STRREF     String }bernehmen
       CB   *R2,@BYT16  Sind 16 Bytes }bergeben ?
       JNE  ERRT2       Wenn nicht, Fehlermeldung
*
       LI   R15,1       Nur 1 Durchgang wenn kein 4.Parameter vorhanden
       SRL  R12,8       Byte in Wort verwandeln
```

```
       CI   R12,4        Sind 4 Parameter vorhanden ?
       JNE  STRIG1       Wenn nicht, sofort mit dem Programm beginnen
*
       INC  R1           4.Parameter
       BL   @TSTGR       Parameter }bernehmen und testen
       DATA 1,768
       MOV  @FAC,R15     Parameter nach R15, gleichzeitig Schleifenz{hler
       JMP  STRIG1       Mit dem eigentlichen Programm beginnen
*
ERRT2  LIMI 2            VDP-Interrupt wieder zulassen
       B    @BARERR      Fehlermeldung ausgeben und Abbruch
*
STRIG1 LI   R1Ø,16       R1Ø=16
       LI   R9,SATZ+1    R9 mit Adresse Stringbuffer's +1 laden
*
STRIG2 CLR  R5           R5=Ø
       MOVB *R9,R5       Ein Byte aus Stringbuffer nach R5
       SWPB R5           Byte in Wort verwandeln
       CI   R5,48        Ist R5 mind. 48 ?
       JLT  ERRT2        Wenn nicht, Fehlermeldung
       CI   R5,7Ø        Ist R5 max. 7Ø ?
       JGT  ERRT2        Wenn nicht, Fehlermeldung
       CI   R5,64        Wenn R5 gr!~er 64, dann sind es Buchstaben
       JGT  STRIG3
       CI   R5,58        Wenn R5 kleiner 58, dann sind es Zahlen
       JLT  STRIG4
       JMP  ERRT2        Wenn alles nicht zutrifft, Fehlermeldung
*
STRIG3 SWPB R5           Wort in Byte verwandeln
       AB   @BYTM55,R5   Buchstabenkonstante addieren
       MOVB R5,*R9+      Byte zur}ck in Stringbuffer und Adresse +1
       JMP  STRIG5
*
STRIG4 SWPB R5           Wort in Byte verwandeln
       AB   @BYTM48,R5   Zahlenkonstante addieren
       MOVB R5,*R9+      Byte zur}ck in Stringbuffer und Adresse +1
*
STRIG5 DEC  R1Ø          Alle Bytes bearbeitet ?
       JNE  STRIG2       Wenn nicht, n{chstes Byte
*
       MOV  @BUFADR,R5   R5 mit aktueller Buffer-Adresse laden
       MOV  R13,RØ       Zeilenwert nach RØ kopieren
       DEC  RØ           RØ=RØ-1
       SLA  RØ,8         RØ=RØ*256
       MOV  R14,R1       Spaltenwert nach R1 kopieren
       DEC  R1           R1=R1-1
       SLA  R1,3         R1=R1*8
       A    R1,RØ        Beide Werte addieren ; Ergebnis ist in RØ
       A    R5,RØ        P.D.T.-Buffer-Adresse dazu
       LI   R4,>18ØØ     R4=6144
       A    R4,R5        Grenzwert in R5 ablegen
*
       LI   R9,MATRIX    R9 mit Variablenadresse laden
       LI   R1Ø,8        Acht Bytes m}~en gebildet werden
       LI   R11,SATZ+1   R11 mit Stringbufferadresse +1 laden
*
STRIG6 CLR  R12          R12=Ø
       MOVB *R11+,R12    Ein Byte nach R12
       MOVB *R11+,R13    Zweites Byte nach R13
       SLA  R12,4        Wert in R12 an sign. Stelle schieben
       SOCB R13,R12      Beide Werte durch log. ODER verkn}pfen
       MOVB R12,*R9+     Fertiges Byte in Variablenbuffer packen
       DEC  R1Ø          Sind acht Bytes verarbeitet ?
       JNE  STRIG6       Wenn nicht, n{chstes Byte
*
STRIG7 LI   R1Ø,8        R1Ø=8
       LI   R9,MATRIX    R9 mit Variablenadresse laden
*
STRIG8 MOVB *R9+,*RØ+    Byte aus Variable in den P.D.T.-Buffer bringen
       DEC  R1Ø          Ein vollst{ndiges Zeichen }bertragen ?
       JNE  STRIG8       Wenn nicht, n{chstes Byte
*
       C    RØ,R5        Ende des P.D.T.-Buffer's erreicht ?
       JEQ  STRIG9       Wenn ja, n{chste Basic-Anweisung
*
       DEC  R15          Sind alle Wiederholungen ausgef}hrt ?
       JNE  STRIG7       Wenn nicht, n{chste Runde
*
STRIG9 LIMI 2            VDP-Interrupt zulassen
       B    @BASIC       N{chste Basic-Anweisung
*
* BL @SQUARE
*
* Dieses Unterprogramm ersetzt die im GPL-Interpreter ent-
* haltene, aber sehr langsame Wurzel-Routine.
*
*
SQUARE MOV  @FAC,R2      Wert nach R2 }bertragen
       C    @D16257,R2   Ist der Wert kleiner als 127^2 ?
       JGT  SQR4         Wenn ja, dann Ø bis 127 ausprobieren
*
       LI   RØ,SQRDA2    RØ mit Werten f}r 128 bis 256 laden
SQR1   C    *RØ+,R2      Ist der Wert aus der Tabelle kleiner ?
       JLT  SQR2         Wenn ja, dann n{chsten Wert
       JMP  SQR3         Wenn nicht, dann ist die richtige Wurzel gefunden
*
SQR2   INCT RØ           Sprung zum n{chsten Oberwert
       JMP  SQR1         Wert untersuchen
*
SQR3   MOV  *RØ,@FAC     Richtigen Wert nach FAC
       JMP  SQR8         Zur}ck zum rufenden Programm
*
SQR4   LI   RØ,SQRDA1    RØ mit Werten f}r Ø bis 127 laden
SQR5   C    *RØ+,R2      Ist der Wert aus der Tabelle kleiner ?
       JLT  SQR6         Wenn ja, dann n{chsten Wert
       JMP  SQR7         Wenn nicht, dann ist die richtige Wurzel gefunden
*
SQR6   INCT RØ           Sprung zum n{chsten Oberwert
       JMP  SQR5         Wert untersuchen
*
SQR7   MOV  *RØ,@FAC     Richtigen Wert nach FAC
*
SQR8   RT                Zur}ck zum rufenden Programm
*
* Ende Segment "BITMAP1S"
*
```

```
* Segment "@BITMAP2S"
*
* ******************
* *                *
* * Unterprogramme *
* *                *
* *     2.Teil     *
* *                *
* ******************
*
* BL @SHUT
*
* Dieses Unterprogramm schlie~t eine Datei
*
SHUT   MOVB @BYT1,R1     Datei-Schlie~-Befehl nach R1
       MOV  @PAB,RØ      In RØ steht die Adresse des Pab im VDP-RAM
       BLWP @VSBW        Byte in's VDP-RAM bringen
       BL   @INOUT       Datei schlie~en
       B    @BASIC       N{chste Basic-Anweisung
*
* BL @GETSTR
*
* }bernimmt den Device-Namen von Basic
*
GETSTR MOV  R11,@BACK1   Returnadresse sichern
       LI   R1,1         Parameternummer in R1 laden
       CLR  RØ           RØ=Ø
       LI   R2,SATZ      R2 mit Adresse des Stringbuffers laden
       MOVB @BYT15,*R2   Maximal 15 Bytes sollen }bernommen werden
       BLWP @STRREF      Device-Namen }bernehmen
       MOV  @PAB,RØ      In PAB+9 mu~ die L{nge des Dateinamen stehen
       A    @DØ9,RØ
       LI   R1,SATZ      In R1 steht die Adresse des Stringbuffers
       MOVB *R1,R2       Aktuelle String{nge nach R2
       CB   @BYTØ,R2     Ist der String null Bytes lang
       JEQ  GETSR1       Wenn ja, Programmabbruch
*
       SRL  R2,8         Byte in Wort verwandeln
       INC  R2           String{nge mu~ mit}bertragen werden
       BLWP @VMBW        String{nge und String an den Pab anh{ngen
       MOV  @BACK1,R11   Returnadresse wiedergewinnen
       RT                Zur}ck zum rufenden Programm
*
GETSR1 B    @BASIC       Eintrittspunkt f}r Programmabbruch bei Nullstring
*
* BL @RESERV
*
* Platz f}r den Pab und den Pab-Buffer im VDP-RAM reservieren
*
RESERV MOVB @BYTØ,@GPLSTA  GPL-Statusbyte l!schen
       LWPI GPLWS          GPL-Arbeitsregister laden
       BLWP @GPLLNK        Platz im VDP-RAM reservieren
       DATA RES
*
       LWPI MYREGI         Eigene Arbeitsregister laden
       MOV  @>831A,@PAB    Die erste freie Adresse in die Variable
       MOV  @PAB,@PABBUF
       A    R5,@PABBUF
       MOV  @PAB,RØ        Adresse des Pab nach RØ
       MOV  RØ,@POINT      Pab-Pointer setzen
       MOV  @BUFADR,R1     Dateidaten-Adresse nach R1
       BLWP @VMBW          Bytes in's VDP-RAM bringen
       INCT RØ             Ab PAB+2 steht die Adresse des Pab-Buffers
       LI   R1,PABBUF      R1 mit der Adresse der Variablen laden
       LI   R2,2           Zwei Bytes sind zu }bertragen
       BLWP @VMBW          Bytes in's VDP-RAM bringen
       RT                  Zur}ck zum rufenden Programm
*
* BL @BRING
*
* Dieses Unterprogramm bringt den neuen Seikosha-String
* in den PAB-Buffer.
*
BRING  MOV  @PABBUF,RØ     RØ mit der Adresse des Pabbuffer's laden
       MOV  R15,R12        Wert von R15 nach R12 kopieren
       SLA  R12,3          R12=Wert von R12 mal 8
       A    R12,RØ         R12=R12+Adresse von PABBUF
       LI   R1,STRINØ      R1 mit Adresse von STRINØ laden
       LI   R2,8           Acht Byte sind zu }bertragen
       BLWP @VMBW          Byte's in das VDP-RAM schreiben
       MOV  @BACK1,R11     Returnadresse wiedergewinnen
       RT                  Zur}ck zum rufenden Programm
*
* BL @DRUCK1
*
* Wenn die erste H{lfte einer Bildschirmzeile f}r den Drucker
* aufbereitet wurde, wird sie durch dieses Unterprogramm
* ausgedruckt und der Drucker f}r die n{chste H{lfte vor-
* bereitet.
*
DRUCK1 MOV  R11,@BACK1      Returnadresse sichern
       LI   R1,>8ØØØ        128 Bytes sollen zum Drucker geschickt werden
       MOV  @PAB5,RØ
       BLWP @VSBW
       BL   @INOUT          Daten zum Drucker schicken
       LI   R1,>Ø4ØØ        Vier Bytes sollen zum Drucker geschickt werden
       MOV  @PAB5,RØ
       BLWP @VSBW
       MOV  @PABBUF,RØ      Daten f}r Druckposition in den PAB-Buffer laden
       LI   R1,MYDAT2
       LI   R2,4            Vier Bytes sind zu }bertragen
       BLWP @VMBW           Bytes in's VDP-RAM schreiben
       BL   @INOUT          Daten zum Drucker schicken
       MOV  @BACK1,R11      Returnadresse wiedergewinnen
       RT                   Zur}ck zum rufenden Programmm
*
* BL @DRUCK2
*
* Wenn die zweite H{lfte einer Bildschirmzeile f}r den Drucker
* aufbereitet wurde, wird sie durch dieses Unterprogramm
* ausgedruckt und der Drucker f}r die n{chste Zeile vor-
* bereitet.
*
DRUCK2 MOV  R11,@BACK1      Returnadresse sichern
       LI   R1,>8ØØØ        128 Bytes sollen zum Drucker geschickt werden
       MOV  @PAB5,RØ
       BLWP @VSBW
       BL   @INOUT          Daten zum Drucker schicken
```

```
       LI   R1,>0500     F}nf Bytes sollen zum Drucker geschickt werden
       MOV  @PAB5,RØ
       BLWP @VSBW
       MOV  @PABBUF,RØ   Daten f}r Druckposition in den PAB-Buffer laden
       LI   R1,MYDAT1
       LI   R2,5         F}nf Bytes sind zu }bertragen
       BLWP @VMBW        Bytes in's VDP-RAM schreiben
       BL   @INOUT       Daten zum Drucker schicken
       MOV  @BACK1,R11   Returnadresse wiedergewinnen
       RT                Zur}ck zum rufenden Programm
*
* BL @DSGNCH
*
* Mit Hilfe dieses Unterprogramms wird aus der Bitmatrix der neue
* Seikosha-String berechnet.
*
DSGNCH MOV  R11,@BACK1   Returnadresse sichern
       CLR  RØ           RØ=Ø
       LI   R1,STRINØ    R1 mit der Adresse von STRINØ laden
       LI   R2,4         4 mal soll die folgende Schleife durchlaufen werden
DSGN1  MOV  RØ,*R1+      Ein Wort l!schen
       DEC  R2           Schleife beendet ?
       JNE  DSGN1        Wenn noch nicht alles gel!scht ist, n{chstes Wort
*
       LI   R9,8         Schleifenz{hler f}r ein Byte
       LI   R12,STRINØ   R12=Adresse von STRINØ
       LI   R15,MATRIX   R15=Adresse der Bitmatrix
       LI   R4,8         Z{hlkonstante f}r aktuelle Bitmatrixadresse
DSGN2  MOV  R15,R1Ø      Bitmatrixadresse nach R1Ø kopieren
       LI   R11,6        Z{hler f}r zu berechnende Bits
       LI   RØ,1         RØ mit Verschiebewert laden
       CLR  R8           R8=Ø
       MOVB *R1Ø,R8      Bit Ø von der Bitmatrix nach R8 kopieren
       A    R4,R1Ø       Aktuelle Bitmatrixadresse um acht erh!hen
DSGN3  MOVB *R1Ø,R13     N{chstes Bit nach R13 bringen
       SWPB R13          Byte in Wort verwandeln
       SLA  R13,Ø        R13 wird nun um Wert von RØ nach links verschoben
*                        --> R13 mal 2^Wert von RØ
       SWPB R13          Wort in Byte verwandeln
       AB   R13,R8       Die einzelnen Bits zu einem Byte aufaddieren
       A    R4,R1Ø       Aktuelle Bitmatrixadresse um acht erh!hen
       INC  RØ           Verschiebewert um eins erh!hen
       DEC  R11          Sind sieben Bits berechnet ?
       JNE  DSGN3        Wenn nicht, n{chstes Bit
*
       AB   @BYT128,R8   Da das achte Bit bei dem Seikosha GP 1ØØ A immer
*                        auf 1 sein mu~, werden hier 128 zu dem Byte
*                        hinzuaddiert.
       MOVB R8,*R12      Fertiges Byte in den String-Buffer packen
       INC  R15          Urspr}ngliche Bitmatrixadresse um eins erh!hen
       INC  R12          String-Bufferadresse um eins erh!hen
       DEC  R9           Sind acht Bytes berechnet ?
       JNE  DSGN2        Nein, also n{chstes Byte berechnen
*
       LWPI MYREGI       Normale Arbeitsregister wieder laden
       BL   @BRING       Seikosha-String in den Pab-Buffer bringen
*
* BL @COPIER
*
* Dieses Unterprogramm zerlegt ein Byte in Bits.
*
COPIER SRL  R5,8         Byte in Wort verwandeln
       SRL  R5,Ø         R5 um Wert von RØ verschieben (dividieren)
       SWPB R5           Wort in Byte verwandeln
       MOVB R5,*R15+     Bit in die Bitmatrix schreiben un Adr. + 1
       SWPB R5           Byte in Wort verwandeln
       SLA  R5,Ø         R5 um Wert von RØ verschieben (multiplizieren)
       SWPB R5           Wort in Byte verwandeln
       SB   R5,R7        Aktuellen Restwert gewinnen
       MOVB R7,R5        Restwert nach R5 kopieren
       DEC  RØ           Ist das Byte zerlegt ?
       JNE  COPIER       Wenn nicht, n{chstes Bit
*
       MOVB R5,*R15+     Bit 7 kann nur noch 1 oder Ø sein
       RT                Zur}ck zum rufenden Programm
*
* BL @INOUT
*
* Dieses Unterprogramm f}hrt die Datei-Operation durch
*
INOUT  MOV  R11,@BACK2   Returnadresse sichern
*
       LWPI MYREG2       2.Satz Arbeitsregister laden
       MOV  @PAB,R6      R6 enth{lt die Adresse der Datei Namenl{nge
       A    @DØ9,R6
       MOV  R6,@PNTR     L{ngenzeiger setzen
       MOVB @BYTØ,@GPLSTA GPL-Statusbyte l!schen
       BLWP @DSRLNK      File-Operation durchf}hren
       DATA 8
       B    @STAND       Operation auf I/O-Fehler untersuchen
*
* BL @STAND
*
* Statusbits auf Dateifehler }berpr}fen
*
STAND  MOV  @PAB,RØ      In PAB+1 sind die Statusbits
       INC  RØ
       BLWP @VSBR        Also Byte holen
       SRL  R1,13        Bit Ø bis Bit 2 m}ssen isoliert werden
       SWPB R1           Wort in Byte verwandeln
       CB   @BYTØ,R1     Sind die Statusbits null ?
       JNE  STAND1       Wenn nicht, Fehlercode festlegen
*
       MOVB @GPLSTA,RØ   GPL-Statusbyte kopieren
       SLA  RØ,2         Bit 2 (COND-BIT) isolieren
       SRL  RØ,15        Bit an Nicht-Sign. Stelle im Nicht-Sign.Byte
       SWPB RØ           Wort in Byte verwandeln
       CB   @BYTØ,RØ     Ist das COND-BIT null ?
       JNE  STAND1       Wenn nicht, I/O-Fehler Ø ausgeben
*
       LWPI MYREGI       Wieder die alten Arbeitsregister laden
       MOV  @BACK2,R11   Returnadresse wiedergewinnen
       RT                Zur}ck zum rufenden Programm
*
STAND1 MOV  @PAB,R2      R2 mit der Adresse des Pab laden
       MOV  R2,@POINT    Adresse in den Pab-Pointer schieben
       MOV  R1,RØ        Statusbit ist gleichzeitig Fehlercode
       BLWP @ERR         Fehler ausgeben und Abbruch
```

```
*
* BL @TRANS
*
* Dieses Unterprogramm kann acht Byte von einer Adresse zu einer anderen
* Adresse transferieren.
*
TRANS  MOV  *R11+,R6       Startadresse nach R6
       MOV  *R11+,R7       Zieladresse nach R7
       LI   RØ,4           Vier Worte sind zu transferieren
TRANS1 MOV  *R6+,*R7+      Wort transferieren
       DEC  RØ             Sind alle vier Worte }bertragen
       JNE  TRANS1         Wenn nicht, n{chstes Wort
*
       RT                  Zur}ck zum rufenden Programmteil
*
* BL @SUBLI1
*
* Dieses Unterprogramm beinhaltet Routinen f}r das Hauptprogramm 'LINE'
*
SUBLI1 MOV  R11,@BACK2     Returnadresse sichern
       BL   @TRANS         Acht-Byte-Block transferieren
       DATA XAFP,FAC
       BLWP @XMLLNK        Gleitkomma-Addition durchf}hren
       DATA FADD
       BL   @TRANS         Acht-Byte-Block transferieren
       DATA FAC,XAFP
       BLWP @XMLLNK        Gleitkomma in Integer verwandeln
       DATA CFI
       MOV  @FAC,@SATZ+2   X-Koordinate nach Satz plus zwei
       MOV  @YA,@SATZ      Y-Koordinate nach Satz
       MOV  @DØ2,@PRMZAL   Zwei Koordinaten sollen bearbeitet werden
       LWPI MYREG2         Zweiten Satz Arbeitsregister laden
       MOVB @BYT1,R7       Die Koordinaten kommen von au~erhalb
       BL   @SETTO         Koordinaten setzen
*
       LWPI MYREGI         Normale Arbeitsregister wieder laden
       MOV  @BACK2,R11     Returnadresse wiedergewinnen
       RT                  Zur}ck zum rufenden Programm
*
* BL @SUBLI2
*
* Dieses Unterprogramm beinhaltet Routinen f}r das Hauptprogramm 'LINE'.
*
SUBLI2 MOV  R11,@BACK2     Returnadresse sichern
       BL   @TRANS         Acht-Byte-Block transferieren
       DATA YAFP,FAC
       BLWP @XMLLNK        Gleitkomma-Addition durchf}hren
       DATA FADD
       BL   @TRANS         Acht-Byte-Block transferieren
       DATA FAC,YAFP
       BLWP @XMLLNK        Gleitkomma in Integer verwandeln
       DATA CFI
       MOV  @FAC,@SATZ     Y-Koordinate nach Satz
       MOV  @XA,@SATZ+2    X-Koordinate nach Satz plus zwei
       MOV  @DØ2,@PRMZAL   Zwei Koordinaten sollen bearbeitet werden
       LWPI MYREG2         Zweiten Satz Arbeitsregister laden
       MOVB @BYT1,R7       Die Koordinaten kommen von au~erhalb
       BL   @SETTO         Koordinaten setzen
*
       LWPI MYREGI         Normale Arbeitsregister wieder laden
       MOV  @BACK2,R11     Returnadresse wiedergewinnen
       RT                  Zur}ck zum rufenden Programm
*
* BL @GTPRM
*
* Numerische Parameter holen
*
GTPRM  BLWP @NUMREF        Parameter holen
       BLWP @XMLLNK        Gleitkomma in Integer verwandeln
       DATA CFI
       RT                  Return zum rufenden Programm
*
* BL @TSTGR
*
* }berpr}ft die }bergebenen Werte auf Zul{~igkeit
*
TSTGR  MOV  *R11+,R2       Untere Grenze
       MOV  *R11+,R3       Obere Grenze
       MOV  R11,@BACK1     Returnadresse sichern
       DEC  R2             Untere Grenze minus eins
       INC  R3             Obere Grenze plus eins
       CLR  RØ             RØ=Ø
       BL   @GTPRM         Parameter }bernehmen
       MOV  @FAC,R4        Parameter nach R4 kopieren
       C    R4,R2          Ist der Wert gr!~er oder gleich UG ?
       JGT  TSTGR1         Wenn ja, obere Grenze testen
*
       B    @BARERR        Wenn nicht, Fehlermeldung
TSTGR1 C    R4,R3          Ist der Wert kleiner oder gleich OG ?
       JLT  TSTGR2         Wenn ja, zur}ck zum rufenden Programm
*
       B    @BARERR        Wenn nicht, Fehlermeldung ausgeben
TSTGR2 MOV  @BACK1,R11     Returnadresse wiedergewinnen
       RT                  Zur}ck zum rufenden Programm
*
* B @BARERR
*
* Unterprogramm Fehlermeldung ausgeben.
*
BARERR LI   RØ,ERR22       RØ mit Fehlercode f}r 'BAD ARGUMENT' laden
       BLWP @ERR
*
* B @BASIC
*
* Dieses Unterprogramm springt in's Basic zur}ck.
*
BASIC  MOVB @BYTØ,@GPLSTA  GPL-Statusbyte l!schen
       LWPI GPLWS          GPL-Arbeitsregister laden
       B    @>ØØ7Ø         N{chste Basic-Anweisung
*
* Ende Segment "BITMAP2S"
*
```

```
* Segment "@BITMAP3S"
*
* ******************
* *                *
* * Hauptprogramme *
* *                *
* *     1.Teil     *
* *                *
* ******************
*
* CALL LINK("CLEAR")
*
* Dieses Programm l|scht den Pat-Buffer und sichert die Standard-Zeichen.
*
CLEAR  LIMI Ø            VDP-Interrupt unterdr}cken
*
       LWPI MYREGI       Eigene Arbeitsregister laden
       LI   RØ,FARBE     RØ mit Adresse des Farbbuffer's laden
       LI   R1,>1212     Vordergrund : schwarz ; Hintergrund : mittelgr}n
       LI   R2,3072      3072 Worte schreiben
CLEAR1 MOV  R1,*RØ+      Wort in den Farbbuffer schreiben und Adresse +2
       DEC  R2           Alle 3072 Worte geschrieben ?
       JNE  CLEAR1       Wenn nicht, n{chstes Wort
*
       LI   RØ,PATT      RØ mit Adresse des P.D.T.-Buffer's laden
       CLR  R1           R1=Ø
       LI   R2,3072      3072 Worte sind zu l|schen
CLEAR2 MOV  R1,*RØ+      Wort in den P.D.T.-Buffer schreiben und Adr. +2
       DEC  R2           Alle 3072 Worte geschrieben ?
       JNE  CLEAR2       Wenn nicht, n{chstes Wort
*
       LI   RØ,1024      Ab 1024 steht der Standard-Zeichensatz
       LI   R1,BUFFER    R1 mit Adresse des Zeichensatz-Buffer's laden
       LI   R2,768       768 Bytes sind zu l|schen
       BLWP @VMBR        Bytes aus VDP-RAM lesen
       LIMI 2            VDP-Interrupt wieder zulassen
       B    @BASIC       N{chste Basic-Anweisung
*
* CALL LINK("SETPIX",Y-Koordinate,X-Koordinate,(bis zu 7 mal))
*
* Dieses Programm setzt Pixel.Dazu wird in 'P' eine 1 abgelegt.
*
SETPIX LIMI Ø            VDP-Interrupt unterdr}cken
*
       LWPI MYREGI       Eigene Arbeitsregister laden
       MOVB @LNKNUM,@ANZAHL  Anzahl der Parameter sichern
       MOVB @BYT1,@P         In 'P' steht eine 1 f}r Pixel setzen
       B    @PIXEL           Pixel setzen
*
* CALL LINK("CLRPIX",Y-Koordinate,X-Koordinate,(bis zu 7 mal))
*
* Dieses Programm l|scht Pixel.Dazu wird in 'P' eine Ø abgelegt.
*
CLRPIX LIMI Ø            VDP-Interrupt unterdr}cken
*
       LWPI MYREGI       Eigene Arbeitsregister laden
       MOVB @LNKNUM,@ANZAHL  Anzahl der Parameter sichern
       MOVB @BYTØ,@P         In 'P' steht eine Ø f}r Pixel l|schen
       B    @PIXEL           Pixel l|schen
*
* CALL LINK("ZEIG",Sekunden)
*
* Nach Aufruf dieses Programm's wird die im Buffer erstellte Grafic
* f}r eine bestimmte Zeit in's VDP-RAM gebracht.
*
ZEIG   LIMI Ø            VDP-Interrupt unterdr}cken
*
       LWPI MYREGI       Eigene Arbeitsregister laden
       MOVB @BYT16,@>83C2 Wirkung der QUIT-Taste aufheben
       LI   R1,1         1.Parameter
       BL   @TSTGR       Parameter holen und testen
       DATA 1,48Ø        UG,OG
       MOV  @FAC,R12     Parameter nach R12 kopieren
       BL   @BITMAP      Videoprozessor auf Bitmap umschalten
       CLR  RØ           RØ=Ø
       MOVB @BYTØ,@WATCH VDP-Interrupt-Timer zur}cksetzen
       LIMI 2            VDP-Interrupt-Timer starten
ZEIG1  CB   @WATCH,@BYT5Ø Sind 5Ø Impulse f}r eine Sekunde abgelaufen ?
       JNE  ZEIG1        Wenn nicht, n{chste Abfrage
*
       MOVB @BYTØ,@WATCH Wenn doch, VDP-Interr.-Timer zur}cksetzen
       INC  RØ           RØ=RØ+1
       C    RØ,R12       Ist die Zeit vorbei
       JNE  ZEIG1        Wenn nicht, n{chster Durchgang
*
       LIMI Ø            VDP-Interrupt unterdr}cken
       B    @ENDE        Videoprozessor wieder auf Grafic-Modus umschalten
*
* CALL LINK("WRITE",Ø oder 1,Zeile,Spalte,"String")
*
* Dieses Programm l{~t ein schreiben in die Grafik zu.
* Erlaubt sind die Zeichen 32 bis 127.
* Wenn erste Parameter 1, dann Text hinein kopieren
* Wenn erste Parameter Ø, dann Hintergrund vorher l|schen
*
WRITE  LIMI Ø            VDP-Interrupt unterdr}cken
*
       LWPI MYREGI       Eigene Arbeitsregister laden
       LI   R1,1         1.Parameter
       BL   @TSTGR       Parameter }bernehmen und testen
       DATA Ø,1
       MOV  @FAC,R14     Parameter in R14 ablegen
       SWPB R14          Wort in Byte verwandeln
       INC  R1           2.Parameter (Zeile)
       BL   @TSTGR       Parameter }bernehmen und testen
       DATA 1,24
       MOV  @FAC,R9      Parameter nach R9
       INC  R1           3.Parameter
       BL   @TSTGR       Parameter }bernehmen und testen
       DATA 1,32
       MOV  @FAC,R1Ø     Parameter nach R1Ø
       CLR  RØ           RØ=Ø
       INC  R1           4.Parameter (String)
       LI   R2,SATZ      R2 mit Adresse des Stringbuffer's laden
       MOVB @BYT32,*R2   Maximal 32 Zeichen }bernehmen
       CLR  R12          R12=Ø
       BLWP @STRREF      String }bernehmen
       LI   R2,SATZ      R2 mit Adresse des Stringbuffer's laden
```

```
       MOVB *R2+,R12    Aktuelle Stringl{nge nach R12 (Schleifenz{hler)
       SWPB R12         Byte in Wort verwandeln
       CI   R12,Ø       Leerstring ?
       JNE  WRITE1      Wenn nicht, dann schreiben
*
       B    @BASIC      Wenn Leerstring, dann zur}ck in's Basic
WRITE1 LI   R5,BUFFER   Adresse des Pattern-Buffer in R5 laden
       MOV  R9,RØ       Zeile nach RØ
       DEC  RØ          RØ=RØ-1     (Zeile - 1)
       SLA  RØ,8        RØ=RØ*256   (Zeilenwert * 256)
       MOV  R1Ø,R1      Spalte nach R1
       DEC  R1          R1=R1-1     (Spalte - 1)
       SLA  R1,3        R1=R1*8     (Spaltenwert * 8)
       A    R1,RØ       RØ=RØ+R1    (Anzeigeadresse  in RØ ablegen)
       LI   R3,PATT     Patt-Buffer-Adresse in R3 laden
       A    R3,RØ       RØ=RØ+R3    (Buffer-Basisadresse hinzu)
WRITE2 CLR  R4          R4=Ø
       MOVB *R2+,R4     Byte aus String-Variablen nach R4 und Adresse +1
       SB   @BYT32,R4   R4=R4-32    (Zeichen Ø bis 31 nicht darstellbar)
       SWPB R4          Byte in Wort verwandeln
       CI   R4,Ø        Ist Zeichen jetzt mind. null ?
       JLT  ERRT3       Wenn kleiner, dann Fehlermeldung
*
       CI   R4,95       Ist Zeichen jetzt max. 95 ?
       JGT  ERRT3       Wenn gr¦~er, dann Fehlermeldung
*
       SLA  R4,3        R4=R4*8     (Zeichenwert * 8)
       A    R5,R4       R4=R4+R5    (Zeichenbuffer-Adresse dazu)
       LI   R1,8        Ein Zeichenstring ist acht Byte lang
WRITE3 CLR  R6          R6=Ø
       CLR  R7          R7=Ø
       MOVB *R4+,R6     Byte aus Zeichenbuffer nach RØ und Adr. +1
       MOVB *RØ,R7      Byte aus Pattbuffer nach R7
       CB   @BYTØ,R14   Soll der Hintergrund gel¦scht werden ?
       JEQ  WRITE4      Ja, also Wert nicht verkn}pfen
*
       SOCB R6,R7       Sign. Bytes durch log. ODER verkn}pfen
       JMP  WRITE5      Byte in Pattbuffer bringen
*
WRITE4 MOVB R6,*RØ+     Hintergrund l¦schen, also Byte sofort in Pattbuffer
       JMP  WRITE6      Jberpr}fe, ob String vollst{ndig
*
WRITE5 MOVB R7,*RØ+     Log. verkn}pften Wert in Pattbuffer bringen
WRITE6 DEC  R1          Ist der Zeichenstring vollst{ndig ?
       JNE  WRITE3      Wenn nicht, n{chstes Byte
*
       DEC  R12         Gesamter Anzeigestring auf dem Bildschirm ?
       JNE  WRITE2      Wenn nicht, n{chstes Zeichen
*
       LIMI 2           VDP-Interrupt zulassen
       B    @BASIC      N{chste Basic-Anweisung
ERRT3  LIMI 2           VDP-interrupt zulassen
       B    @BARERR     Fehlermeldung und Abbruch
*
* CALL LINK("CHAR",Zeile,Spalte,"String",Wiederholungen)
*
* Dieses Programm erlaubt es, ein Zeichen durch einen String
* umzudefinieren (genau wie CALL CHAR in Basic).Es sind maximal 768
* Wiederholungen erlaubt.
```

```
*
CHAR   LIMI Ø           VDP-Interrupt unterdr}cken
*
       LWPI MYREGI      Eigene Arbeitsregister laden
       LI   R1,PATT     R1 mit P.D.T.-Buffer-Adresse laden
       MOV  R1,@BUFADR  Adresse sichern
       BL   @STRING     In's eigentliche Programm springen
*
* CALL LINK("COLOR",Vordergrundfarbe,Hintergrundfarbe)
*                  ,Zeile,Spalte,"String",Wiederholungen)
*
* Dieses Programm erlaubt es die Vordergrund- und Hintergrundfarbe
* des ganzen Bildschirm's auf einmal zu {ndern oder aber jede Zeichen-
* position farblich mittels eines Strings zu definieren.
*
COLOR  LIMI Ø           VDP-Interrupt unterdr}cken
*
       LWPI MYREGI      Eigene Arbeitsregister laden
       MOVB @LNKNUM,RØ  Anzahl der Parameter sichern
       CB   @BYT2,RØ    Sind zwei Parameter }bergeben worden ?
       JEQ  COLOR1      Wenn ja, dann Farbbuffer {ndern
*
       LI   R1,FARBE    R1 mit Adresse des Farbbuffer's laden
       MOV  R1,@BUFADR  Adresse zwischenspeichern
       BL   @STRING     String in den Farbbuffer bringen
COLOR1 LI   R1,1        1.Parameter
       BL   @TSTGR      Parameter }bernehmen und testen
       DATA 1,16
       MOV  @FAC,R13    Vordergrundfarbe nach R13
       INC  R1          2.Parameter
       BL   @TSTGR      Parameter }bernehmen und testen
       DATA 1,16
       MOV  @FAC,R14    Hintergrundfarbe nach R14
       DEC  R13         V-Farbe -1
       DEC  R14         H-Farbe -1
       SWPB R13         Wort in Byte verwandeln
       SWPB R14         Wort in Byte verwandeln
       SLA  R13,4       V-Farbe um vier Bits nach links schieben
       SOCB R14,R13     Beide Werte mit log. ODER verkn}pfen
       CLR  R1          R1=Ø
       MOVB R13,R1      Fertiges Byte nach R1
       LI   RØ,FARBE    RØ mit der Adresse des Farbbuffer's laden
       LI   R2,6144     6144 Bytes sind zu {ndern
COLOR2 MOVB R1,*RØ+     Byte in den Farbbuffer und Adresse +1
       DEC  R2          Sind alle Bytes ge{ndert ?
       JNE  COLOR2      Wenn nicht, n{chstes Byte
*
       LI   R13,1       R13=1
       SWPB R13         Wort in Byte verwandeln
       SLA  R13,4       R13=R13*16
       SOCB R14,R13     Beide Werte mit log. ODER verkn}pfen
       LI   RØ,VREG1+15 Zieladresse laden
       MOVB R13,*RØ     Berechneten Wert im Ziel speichern
       LIMI 2           VDP-Interrupt wieder zulassen
       B    @BASIC
*
* CALL LINK("HCOPY")
*
```

```
* Dieses Programm erlaubt es, eine Bildschirm-Hardcopy auf einem
* Seikosha GP 100 A auszugeben.
*
HCOPY  LWPI MYREGI      Eigene Arbeitsregister laden
       LI   R15,PDATA2   Adresse der aktuellen PAB-Daten laden
       MOV  R15,@BUFADR  Adresse sichern
       MOV  @D152,@>830C 152 Bytes sollen im VDP-RAM reserviert werden
       LI   R2,16        Die Pabdaten sind sechzehn Bytes lang
       MOV  @D16,R5
       BL   @RESERV
       MOV  @PABBUF,@PRMZAL Adresse des String-Buffer's im VDP-RAM sichern
       A    @D128,@PRMZAL  Der Buffer liegt 128 Bytes hinter dem PAB-Buffer
       MOV  @PAB,R0     Adresse des Pab nach R0
       A    @D05,R0     F}nf hinzuaddieren
       MOV  R0,@PAB5    Adresse in den Buffer schieben
       BL   @INOUT      Datei er!ffnen
       MOVB @WRITEE,R1  Ausgabe-Befehl in's PAB bringen
       MOV  @PAB,R0
       BLWP @VSBW
       BL   @INOUT      Datei auf Ausgabe {ndern
       LI   R1,>0100    Ein Byte zum Drucker schicken
       MOV  @PAB5,R0
       BLWP @VSBW
       MOV  @PABBUF,R0
       LI   R1,>0800    Grafikcode in den PAB-Buffer schreiben
       BLWP @VSBW
       BL   @INOUT      Drucker auf Grafik umstellen
       LI   R1,>0500    F}nf Bytes zum Drucker schicken
       MOV  @PAB5,R0
       BLWP @VSBW
       MOV  @PABBUF,R0  Die 1. H{lfte ab Position 68 drucken
       LI   R1,MYDAT1
       LI   R2,5
       BLWP @VMBW
       BL   @INOUT      Daten zum Drucker schicken
       LI   R10,28      Druckzeilen-Z{hler
       LI   R8,ZLDATA   Adresse der Zeilenbearbeitungsdaten
HCOPY1 LI   R9,2        Eine Druckzeile besteht aus zwei H{lften
       MOV  *R8+,R13    1.Adresse nach R13
       MOV  *R8+,R14    2.Adresse nach R14
       MOV  *R8+,@HELP1 Derzeitige Variablen-Adresse nach HELP1
       MOVB *R8+,@HELP2 1.Schleifenz{hler nach HELP2
       MOVB *R8+,@HELP3 2.Schleifenz{hler nach HELP3
       LI   R11,PATT    Adresse des Pat-Buffer's laden
       A    R11,R13     Konstante hinzuaddieren
       A    R11,R14
HCOPY2 CLR  R15         R15=0
HCOPY3 MOV  R13,R1      CPU-Adresse nach R1
       A    @D08,R13    Adresse um Stringl{nge erh!hen
       MOV  @PRMZAL,R0  VDP-Bufferadresse nach R0
       LI   R2,8        Acht Bytes sind zu }bertragen
       BLWP @VMBW       String sichern
       LI   R1,STRIN1   CPU-Adresse des String laden
       BLWP @VMBR       String in's Ziel bringen
       MOV  R14,R1      CPU-Adresse nach R1
       A    @D08,R14    Adresse um Stringl{nge erh!hen
       BLWP @VMBW       String sichern
       LI   R1,STRIN2   CPU-Adresse des String laden
       BLWP @VMBR       String in's Ziel bringen
*
       LWPI MYREG2      2.Satz Arbeitsregister laden
       LI   R15,MATRIX  R15 mit der Adresse der Bitmatrix laden
       LI   R2,56       56 Bytes sind zu l!schen
HCOPY4 MOVB @BYT0,*R15+ Byte l!schen und Adresse plus eins
       DEC  R2          Alle Bytes gel!scht ?
       JNE  HCOPY4      Wenn nicht, n{chstes Byte
*
       CLR  R1          R1=0
       LI   R15,MATRIX  Bitmatrixadresse erneuern
       MOV  @HELP1,R14  Aktuelle STRIN1- oder STRIN2-Adresse gewinnen
       LI   R13,STRIN2  R13 mit STRIN2-Adresse laden
       MOVB @HELP2,R9   Ersten SChleifenz{hlerwert laden
       MOVB R9,R9       Soll die Schleife ausgef}hrt werden ?
       JEQ  HCOPY6      Nein, also n{chste Schleife
*
       SWPB R9          Ja, also Wort in Byte verwandeln
       MOVB R1,R9       MS-Byte von R9 l!schen
HCOPY5 LI   R0,7        R0 mit Verschiebewert laden
       MOVB *R14+,R5    Byte nach R5 kopieren
       MOVB R5,R7       Byte in R7 sichern
       BL   @COPIER     Byte in Bitmatrix eintragen
       DEC  R9          Schleifenz{hler um eins erniedrigen
       JNE  HCOPY5      Wenn Schleife noch nicht fertig, n{chstes Byte
*
HCOPY6 MOVB @HELP3,R9   Zweiten Schleifenz{hlerwert laden
       MOVB R9,R9       Soll die Schleife ausgef}hrt werden ?
       JEQ  HCOPY8      Nein, also Bitmatrix auswerten
*
       SWPB R9          Ja, also Wort in Byte verwandeln
       MOVB R1,R9       MS-Byte von R9 l!schen
HCOPY7 LI   R0,7        R0 mit Verschiebewert laden
       MOVB *R13+,R5    Byte nach R5 kopieren
       MOVB R5,R7       Byte in R7 sichern
       BL   @COPIER     Byte in Bitmatrix eintragen
       DEC  R9          Schleifenz{hler um eins erniedrigen
       JNE  HCOPY7      Wenn Schleife noch nicht fertig, n{chstes Byte
*
HCOPY8 BL   @DSGNCH     Bitmatrix auswerten
       INC  R15         Spaltenz{hler um eins erh!hen
       CI   R15,16      Eine Zeilenh{lfte ausgewertet ?
       JNE  HCOPY3      Nein, also n{chste Spalte
*
       DEC  R9          Eine komplette Zeile ausgewertet ?
       JNE  HCOPY9      Wenn nicht, erste H{lfte drucken
*
       BL   @DRUCK2     Sonst zweite H{lfte drucken
       JMP  HCOPYA      N{chste Zeile auswerten
*
HCOPY9 BL   @DRUCK1     Erste H{lfte drucken
       JMP  HCOPY2      Zweite h{lfte auswerten
*
HCOPYA DEC  R10         Druckzeilenz{hler um eins erniedrigen
       JNE  HCOPY1      Noch nicht alle Zeilen gedruckt
*
       LI   R1,>0500    Alle Zeilen gedruckt, f}nf Bytes zum Drucker
       MOV  @PAB5,R0
       BLWP @VSBW
       MOV  @PABBUF,R0  Daten f}r Zeilenvorschub laden
```

```
       LI   R1,MYDAT3
       LI   R2,5
       BLWP @VMBW
       BL   @INOUT       Zeilenvorschub ausf}hren
       LI   R1,>0100     Ein Byte zum Drucker schicken
       MOV  @PAB5,R0
       BLWP @VSBW
       MOV  @PABBUF,R0
       LI   R1,>0F00     Textmoduscode in den PAB-Buffer schreiben
       BLWP @VSBW
       BL   @INOUT       Drucker wieder auf Standardmodus umschalten
       CLR  R1           Aus PAB-Buffer kein Byte mehr zum Drucker schicken
       MOV  @PAB5,R0
       BLWP @VSBW
       B    @SHUT        Datei schlie~en und n{chste Basic-Anweisung
*
* CALL LINK("SAVE","DSK"(1-3)"."(NAME))
*
* Dieses Programm erlaubt es, die Grafic im Variable 80 - Format auf
* Diskette abzuspeichern.
*
SAVE   LWPI MYREGI       Eigene Arbeitsregister laden
       LI   R15,PDATA      Adresse der aktuellen PAB-Daten laden
       MOV  R15,@BUFADR    Adresse sichern
       MOV  @D105,@>830C   105 Bytes sollen im VDP-RAM reserviert werden
       LI   R2,10          Die Pabdaten sind zehn Bytes lang
       MOV  @D25,R5
       BL   @RESERV        Platz f}r den Pab-Buffer im VDP-RAM reservieren
       LI   R1,>1200     Display/Variable/Output
       MOV  @PAB,R0      In R0 steht die Adresse von Pab
       INC  R0           R0=R0+1
       BLWP @VSBW        Byte in's VDP-RAM bringen
       BL   @GETSTR      Datei-Namen }bernehmen und in's VDP-RAM bringen
       BL   @INOUT       Datei er!ffnen
       LI   R1,>0300     Byte f}r Write
       MOV  @PAB,R0      Pab-Adresse laden
       BLWP @VSBW        Byte in's VDP-RAM bringen
       BL   @INOUT       Datei-Operation durchf}hren
       LI   R14,PATT     R14 mit Adresse des P.D.T.-Buffer's laden
       MOV  @PAB,R0      In PAB+5 steht die Anzahl der Zeichen f}r Write
       A    @D05,R0
       LI   R1,80        80 Bytes sollen bei Write gespeichert werden
       SWPB R1           Wort in Byte verwandeln
       BLWP @VSBW        Ein Byte in's VDP-RAM schreiben
       MOV  @PABBUF,R0   In R0 steht die Adresse des Pab-Buffers im VDP-RAM
       LI   R2,80        80 Bytes m}~en jedesmal in den Pab-Buffer
       LI   R12,153      153 mal 80 Bytes m}~en gespeichert werden
SAVE1  MOV  R14,R1       Aktuelle Buffer-Adresse nach R1
       BLWP @VMBW        Bytes in den Pab-Buffer bringen
       A    @D80,R14     Adressenkonstante hinzuaddieren
       BL   @INOUT       Datei-Operation durchf}hren
       DEC  R12          Sind die 153 Durchg{nge ausgef}hrt ?
       JNE  SAVE1        Wenn nicht, n{chster Durchgang
*
       MOV  @PAB,R0      Jetzt m}~en noch 64 Bytes gespeichert werden
       A    @D05,R0
       LI   R1,48        R1=48
       SWPB R1           Wort in Byte verwandeln
       BLWP @VSBW        Byte in's VDP-RAM bringen
       MOV  @PABBUF,R0   In R0 steht die Adresse des Pab-Buffer's
       MOV  R14,R1       Wieder die aktuelle Buffer-Adresse nach R1
       LI   R2,48        Die restlichen 48 Bytes in den Pab-Buffer bringen
       BLWP @VMBW        Bytes in's VDP-RAM bringen
       BL   @INOUT       Datei-Operation durchf}hren
       B    @SHUT        Datei schlie~en und zur}ck in's Basic
*
* CALL LINK("OLD","DSK"(1-3)"."(NAME))
*
* Dieses Programm erlaubt es, die auf Diskette gespeicherten Grafiken
* zu laden.
*
OLD    LWPI MYREGI       Eigene Arbeitsregister laden
       LI   R15,PDATA      Adresse der aktuellen PAB-Daten laden
       MOV  R15,@BUFADR    Adresse sichern
       MOV  @D105,@>830C   105 Bytes sollen im VDP-RAM reserviert werden
       LI   R2,10          Die Pabdaten sind zehn Bytes lang
       MOV  @D25,R5
       BL   @RESERV        Platz f}r den Pab-Buffer im VDP-RAM reservieren
       LI   R1,>1400     Display/Variable/Input
       MOV  @PAB,R0      In R0 steht die Adresse des Pab
       INC  R0           R0=R0+1
       BLWP @VSBW        Byte in's VDP-RAM bringen
       BL   @GETSTR      Datei-Namen }bernehmen und in's VDP-RAM bringen
       BL   @INOUT       Datei er!ffnen
       LI   R1,>0200     Byte f}r Read
       MOV  @PAB,R0      Pab-Adresse laden
       BLWP @VSBW        Byte in's VDP-RAM bringen
       BL   @INOUT       Datei-Operation durchf}hren
       LI   R14,PATT     R14 mit Adresse des P.D.T.-Buffer's laden
       MOV  @PABBUF,R0   In R0 steht die Adresse des P.D.T.-Buffer's
       LI   R2,80        80 Bytes m}~en aus dem Pab-Buffer gelesen werden
       LI   R12,153      153 mal 80 Bytes m}~en geladen werden
OLD1   BL   @INOUT       Datei-Operation durchf}hren
       MOV  R14,R1       Aktuelle Buffer-Adresse nach R1
       BLWP @VMBR        Bytes aus dem Pab-Buffer sichern
       A    @D80,R14     Adressenkonstante hinzuaddieren
       DEC  R12          Sind die 153 Durchg{nge durchgef}hrt ?
       JNE  OLD1         Wenn nicht, n{chster Durchgang
*
       BL   @INOUT       Datei-Operation durchf}hren
       MOV  R14,R1       Wieder aktuelle Buffer-Adresse nach R1
       LI   R2,48        Die restlichen 48 Bytes aus dem Pab-Buffer lesen
       MOV  @PABBUF,R0   In R0 steht die Adresse des Pab-Buffer's
       BLWP @VMBR        Bytes aus dem VDP-RAM lesen
       B    @SHUT
*
* Ende Segment "@BITMAP3S"
*
```

```
* Segment "@BITMAP4S"
*
* ******************
* *                *
* * Hauptprogramme *
* *                *
* *     2.Teil     *
* *                *
* ******************
*
* CALL LINK("CKREIS",Y,X,R)
*
* Dieses Programm erlaubt es, durch Jbergabe der Mittelpunkt-Koordinaten
* und des Radius Kreise zu l|schen.
*
CKREIS LIMI Ø              VDP-Interrupt unterdr}cken
*
       LWPI MYREGI         Eigene Arbeitsregister laden
       MOVB @BYTØ,@P       Pixel l|schen
       JMP  KREIS1         Sprung in's Hauptprogramm
*
* CALL LINK("KREIS",Y,X,R)
*
* Dieses Programm erlaubt es, durch Jbergabe der Mittelpunkt-Koordinaten
* und des Radius Kreise zu erzeugen.
*
KREIS  LIMI Ø              VDP-Interrupt unterdr}cken
*
       LWPI MYREGI         Eigene Arbeitsregister laden
       MOVB @BYT1,@P       Pixel setzen
KREIS1 LI   R1,1           1.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,192
       MOV  @FAC,@Y        Parameter ist Y-Koordinate
       INC  R1             2.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,256
       MOV  @FAC,@X        Parameter ist X-Koordinate
       INC  R1             3.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,255
       MOV  @FAC,R9        Parameter ist Radius in Bildschirmpixel
       MOV  @FAC,R5        Radius ist jetzt in R5 und R9
       MPY  R9,R5          Den Radius quadrieren
       MOV  R6,@RQ         Ergebnis in RQ sichern
       CLR  R7             R7=Ø
KREIS2 MOV  R7,R5          R7=R5 , R7 ist gleichzeitig Schleifenz{hler
       MPY  R7,R5          Schleifenz{hler quadrieren
       MOV  @RQ,R8         R8=RQ
       S    R6,R8          R8=R8-R6
       MOV  R8,@FAC        R8 nach FAC bringen
*
       LWPI MYREG2         2.Satz Arbeitsregister laden
       BL   @SQUARE        Wurzel aus FAC und Ergebnis nach FAC
*
       LWPI MYREGI         Eigene Arbeitsregister wieder laden
       MOV  @FAC,R8        Ergebnis zur}ck nach R8
       MOV  @X,R11         R11=X-Koordinate
       S    R7,R11         R11=R11-R7
       MOV  R11,@SATZ+2    Ergebnis an richtige Position bringen
       MOV  R11,@SATZ+6
       MOV  @Y,R11         R11=Y-Koordinate
       A    R8,R11         R11=R11+R8
       MOV  R11,@SATZ
       MOV  R11,@SATZ+8
       MOV  @X,R11         R11=X-Koordinate
       A    R7,R11         R11=R11+R7
       MOV  R11,@SATZ+1Ø
       MOV  R11,@SATZ+14
       MOV  @Y,R11         R11=Y-Koordinate
       S    R8,R11         R11=R11-R8
       MOV  R11,@SATZ+4
       MOV  R11,@SATZ+12
       MOV  @X,R11         R11=X-Koordinate
       S    R8,R11         R11=R11-R8
       MOV  R11,@SATZ+18
       MOV  R11,@SATZ+22
       MOV  @Y,R11         R11=Y-Koordinate
       A    R7,R11         R11=R11+R7
       MOV  R11,@SATZ+16
       MOV  R11,@SATZ+24
       MOV  @X,R11         R11=X-Koordinate
       A    R8,R11         R11=R11+R8
       MOV  R11,@SATZ+26
       MOV  R11,@SATZ+3Ø
       MOV  @Y,R11         R11=Y-Koordinate
       S    R7,R11         R11=R11-R7
       MOV  R11,@SATZ+2Ø
       MOV  R11,@SATZ+28
*
       LWPI MYREG2         Zweiten Satz Arbeitsregister laden
       MOV  @D32,@PRMZAL   32 Koordinaten sollen bearbeitet werden
       MOVB @BYT1,R7       Die Koordinaten kommen von au~erhalb
       BL   @SETTO         Koordinaten setzen
*
       LWPI MYREGI         Normale Arbeitsregister wieder laden
       INC  R7             Schleifenz{hler um eins erh|hen
       DEC  R9             Wert des Radius um eins verringern
       JNE  KREIS2         Wenn der Radius noch nicht null, n{chster Durchgang
*
       LIMI 2              VDP-Interrupt wieder zulassen
       B    @BASIC         N{chste Basic-Anweisung
*
* CALL LINK("CWLINE",Y,X,L{nge,Winkel zur Horizontalen)
*
* Dieses Programm erlaubt es, durch Angabe eines Punktes, der L{nge der
* Linie und eines Winkels zur Horizontalen Linien zu l|schen.
* (Positive Winkel drehen im Urzeigersinn)
*
CWLINE LIMI Ø              VDP-Interrupt unterdr}cken
*
       LWPI MYREGI         Eigene Arbeitsregister laden
       MOVB @BYTØ,@P       Pixel l|schen
       JMP  LINE1          Sprung in's Hauptprogramm
*
* CALL LINK("WLINE",Y,X,L{nge,Winkel zur Horizontalen)
*
* Dieses Programm erlaubt es, durch Angabe eines Punktes, der L{nge der
```

```
* Linie und eines Winkels zur Horizontalen Linien zu erzeugen.
* (Positive Winkel drehen im Urzeigersinn)
*
WLINE  LIMI Ø              VDP-Interrupt unterdr}cken
*
       LWPI MYREGI         Eigene Arbeitsregister laden
       MOVB @BYT1,@P       Pixel setzen
LINE1  LI   R1,1           1.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,192
       MOV  @FAC,@YA       Y-Koordinate des Anfangspunktes nach YA
       INC  R1             2.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,256
       MOV  @FAC,@XA       X-Koordinate des Anfangspunktes nach XA
       INC  R1             3.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,32Ø
       BLWP @XMLLNK        L{nge der Linie in Gleitkomma verwandeln
       DATA CIF
       BL   @TRANS         Gleitkomma-Wert sichern
       DATA FAC,FPBUF1
       INC  R1             4.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA Ø,359
       BLWP @XMLLNK        Winkel in Gleitkomma verwandeln
       DATA CIF
       BL   @TRANS         Gleitkomma-Wert sichern
       DATA FAC,ARG
       BL   @TRANS         Divisions-Wert nach FAC
       DATA DIVISO,FAC
       BLWP @XMLLNK        Winkel in Radiant umwandeln
       DATA FDIV
       BL   @TRANS         Radiant-Winkel sichern
       DATA FAC,WINKEL
       MOVB @BYTØ,@GPLSTA  GPL-Statusbyte l!schen
       LWPI GPLWS          GPL-Arbeitsregister laden
       BLWP @GPLLNK        Cosinus des Winkels berechnen
       DATA COS
*
       LWPI MYREGI         Normale Arbeitsregister wieder laden
       BL   @TRANS         L{nge der Linie nach ARG bringen
       DATA FPBUF1,ARG
       BLWP @XMLLNK        Cosinus des Winkels und L{nge multiplizieren
       DATA FMULT
       BLWP @XMLLNK        Ergebnis in Integer verwandeln
       DATA CFI
       MOV  @FAC,@XE       Ergebnis ist X-Koordinate des Endpunktes
       BL   @TRANS         Winkel wieder nach FAC bringen
       DATA WINKEL,FAC
       MOVB @BYTØ,@GPLSTA  GPL-Statusbyte l!schen
       LWPI GPLWS          GPL-Arbeitsregister laden
       BLWP @GPLLNK        Sinus des Winkels berechnen
       DATA SIN
*
       LWPI MYREGI         Normale Arbeitsregister wieder laden
       BL   @TRANS         L{nge der Linie nach ARG
       DATA FPBUF1,ARG
       BLWP @XMLLNK        Sinus des Winkels und l{nge multiplizieren
       DATA FMULT
       BLWP @XMLLNK        Ergebnis in Integer umwandeln
       DATA CFI
       MOV  @FAC,@YE       Ergebnis ist Y-Koordinate des Endpunktes
       A    @XA,@XE        X-Koordinate des Endpunktes vervollst{ndigen
       A    @YA,@YE        Y-Koordinate des Endpunktes vervollst{ndigen
       JMP  LINE3          Sprung in's Hauptprogramm
*
* CALL LINK("CLINE",Y1,X1,Y2,X2)
*
* Dieses Programm erlaubt es, zwei angegebene Punkte mittels einer Linie
* zu verbinden. Bei diesem Programm wird die Linie gel!scht.
*
CLINE  LIMI Ø              VDP-Interrupt unterdr}cken
       LWPI MYREGI         Eigene Arbeitsregister laden
       MOVB @BYTØ,@P       Pixel l!schen
       JMP  LINE2          Sprung zur Koordinaten-}bernahme
*
* CALL LINK("LINE",Y1,X1,Y2,X2)
*
* Dieses Programm erlaubt es, zwei angegebene Punkte mittels einer Linie
* zu verbinden. Bei diesem Programm wird die Linie erzeugt.
*
LINE   LIMI Ø              VDP-Interrupt unterdr}cken
*
       LWPI MYREGI         Eigene Arbeitsregister laden
       MOVB @BYT1,@P       Pixel setzen
LINE2  LI   R1,1           1.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,192
       MOV  @FAC,@YA       Y-Koordinate des Anfangspunktes sichern
       INC  R1             2.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,256
       MOV  @FAC,@XA       X-Koordinate des Anfangspunktes sichern
       INC  R1             3.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,192
       MOV  @FAC,@YE       Y-Koordinate des Endpunktes sichern
       INC  R1             4.Parameter
       BL   @TSTGR         Parameter }bernehmen und testen
       DATA 1,256
       MOV  @FAC,@XE       X-Koordinate des Endpunktes sichern
LINE3  MOV  @YA,@FAC       Y-Anfangskoordinate nach FAC
       BLWP @XMLLNK        Wert in Gleitkomma umwandeln
       DATA CIF
       BL   @TRANS         Gleitkomma-Wert sichern
       DATA FAC,YAFP
       MOV  @XA,@FAC       X-Anfangskoordinate nach FAC
       BLWP @XMLLNK        Wert in Gleitkomma umwandeln
       DATA CIF
       BL   @TRANS         Gleitkomma-Wert sichern
       DATA FAC,XAFP
       MOV  @YE,@FAC       Y-Endkoordinate nach FAC
       BLWP @XMLLNK        Wert in Gleitkomma umwandeln
       DATA CIF
       BL   @TRANS         Gleitkomma-Wert sichern
       DATA FAC,YEFP
       MOV  @XE,@FAC       X-Endkoordinate nach FAC
```

```
       BLWP @XMLLNK          Wert in Gleitkomma umwandeln
       DATA CIF
       BL   @TRANS           Gleitkomma-Wert sichern
       DATA FAC,XEFP
*
       LWPI MYREG2           Zweiten Satz Arbeitsregister laden
       MOV  @DØ2,@PRMZAL     Zwei Koordinaten sollen bearbeitet werden
       MOVB @BYT1,R7         Die Koordinaten kommen von au~erhalb
       MOV  @YA,@SATZ        Y-Anfangskoordinate nach SATZ
       MOV  @XA,@SATZ+2      X-Anfangskoordinate nach SATZ plus zwei
       BL   @SETTO           Koordinaten bearbeiten
*
       LWPI MYREGI           Normale Arbeitsregister wieder laden
       MOV  @XA,R2           Hier werden die Anfangs- und Endkkordinaten in
       MOV  @YA,R3           Registern gesichert. Zur Jbersichtlichkeit be-
       MOV  @XE,R4           nutze ich jetzt nur noch die Variablen zur Er-
       MOV  @YE,R5           kl{rung.
       C    R2,R4            Ist XA=XE ?
       JNE  LINE4            Wenn nicht, Sprung
*
       C    R3,R5            Wenn doch, ist auch YA=YE ?
       JNE  LINE4            Wenn nicht, Sprung
*
       LIMI 2                Wenn doch, VDP-Interrupt wieder zulassen
       B    @BASIC           und n{chste Basic-Anweisung
LINE4  S    R2,R4            R4=XE-XA
       S    R3,R5            R5=YE-YA
       MOV  R4,R13           R13=XE-XA
       MOV  R5,R15           R15=YE-YA
       ABS  R4               ABS(XE-XA)
       ABS  R5               ABS(YE-YA)
       C    R4,R5            Ist R4>R5 ?
       JGT  LINE5            Wenn ja, Sprung
*
       JMP  LINE6            Sonst Sprung
*
LINE5  B    @LINEB
LINE6  MOV  R13,@FAC         XE-XA steht jetzt in FAC
       BLWP @XMLLNK          Wert in Gleitkomma umwandeln
       DATA CIF
       BL   @TRANS           Den FAC-Wert nach ARG kopieren
       DATA FAC,ARG
       MOV  R5,@FAC          ABS(YE-YA) nach FAC
       BLWP @XMLLNK          Wert in Gleitkomma-Wert umwandeln
       DATA CIF
       BLWP @XMLLNK          Gleitkomma-Division: ARG/FAC
       DATA FDIV
       BL   @TRANS           Ergebnis sichern
       DATA FAC,FPBUF1
       MOV  @YA,R8           R8=YA
       MOV  @YE,R9           R9=YE
       S    R9,R8            R8=YA-YE
       ABS  R8               ABS(YA-YE)
       INC  R8               R8=R8+1
       CI   R15,Ø            Ist R15=Ø ?
       JGT  LINE9            Wenn R15 gr!~er ist, Sprung
*
LINE7  DEC  @YA              YA=YA-1
       DEC  R8               Ist R8=Ø ?
       JNE  LINE8            Wenn nicht, Sprung
*
       LIMI 2                Wenn doch, VDP-Interrupt wieder zulassen
       B    @BASIC           N{chste Basic-Anweisung
LINE8  BL   @TRANS           Gleitkomma-Wert nach ARG transferieren
       DATA FPBUF1,ARG
       BL   @SUBLI1          Berechnung durchf}hren und Koordinaten setzen
       JMP  LINE7            N{chste Schleife durchf}hren
*
LINE9  INC  @YA              YA=YA+1
       DEC  R8               Ist R8=Ø ?
       JNE  LINEA            Wenn nicht, Sprung
*
       LIMI 2                Wenn doch, VDP-Interrupt wieder zulassen
       B    @BASIC           N{chste Basic-Anweisung
LINEA  BL   @TRANS           Gleitkomma-Wert nach ARG transferieren
       DATA FPBUF1,ARG
       BL   @SUBLI1          Berechnung durchf}hren und Koordinaten setzen
       JMP  LINE9            N{chste Schleife durchf}hren
*
LINEB  MOV  R15,@FAC         YE-YA steht jetzt in FAC
       BLWP @XMLLNK          Wert in Gleitkomma umwandeln
       DATA CIF
       BL   @TRANS           Den FAC-Wert nach ARG kopieren
       DATA FAC,ARG
       MOV  R4,@FAC          ABS(XE-XA) nach FAC
       BLWP @XMLLNK          Wert in Gleitkomma-Wert umwandeln
       DATA CIF
       BLWP @XMLLNK          Gleitkomma-Division: ARG/FAC
       DATA FDIV
       BL   @TRANS           Ergebnis sichern
       DATA FAC,FPBUF1
       MOV  @XA,R8           R8=XA
       MOV  @XE,R9           R9=XE
       S    R9,R8            R8=XA-XE
       ABS  R8               ABS(XA-XE)
       INC  R8               R8=R8+1
       CI   R13,Ø            Ist R13=Ø ?
       JGT  LINEE            Wenn R13 gr!~er ist, Sprung
*
LINEC  DEC  @XA              XA=XA-1
       DEC  R8               Ist R8=Ø ?
       JNE  LINED            Wenn nicht, Sprung
*
       LIMI 2                Wenn doch, VDP-Interrupt wieder zulassen
       B    @BASIC           N{chste Basic-Anweisung
LINED  BL   @TRANS           Gleitkomma-Wert nach ARG transferieren
       DATA FPBUF1,ARG
       BL   @SUBLI2          Berechnung durchf}hren und Koordinaten setzen
       JMP  LINEC            N{chste Schleife durchf}hren
*
LINEE  INC  @XA              XA=XA+1
       DEC  R8               Ist R8=Ø ?
       JNE  LINEF            Wenn nicht, Sprung
*
       LIMI 2                Wenn doch, VDP-Interrupt wieder zulassen
       B    @BASIC           N{chste Basic-Anweisung
LINEF  BL   @TRANS           Gleitkomma-Wert nach ARG transferieren
       DATA FPBUF1,ARG
```

```
BL   @SUBLI2        Berechnung durchführen und Koordinaten setzen
JMP  LINEE          Nächste Schleife durchführen

*** Ende Segment "@BITMAP4S"
```

Fortsetzung von S. 41

BIT MAP MODE

hen alle ASCII-Zeichen von 32 bis 126 einschließlich.

CALL LINK ("CHAR", Zeile, Spalte, String, Wiederholungen (Optional))–->
Dieser Befehl entspricht dem von Basic her bekannten Befehl CALL CHAR, jedoch wirkt er nicht auf ein bestimmtes Zeichen, sondern auf eine Bildschirmposition. Der String muß 16 Zeichen lang sein. Parameter vier ist optional, d.h. daß der definierte String ab der angegebenen Position bis zu 768mal auftauchen kann. Wird der vierte Parameter nicht angegeben, so erscheint der selbstdefinierte String nur einmal auf dem Bildschirm.

CALL LINK ("COLOR", Zeile, Spalte, String, Wiederholungen (Optional))–->
Dieser Befehl entspricht im wesentlichen dem Befehl CALL LINK ("CHAR",...), jedoch haben Sie hiermit die Möglichkeit, die Farben der Grafik zu beeinflussen. Jeweils acht Pixel können eine Vordergrund- und eine Hintergrundfarbe annehmen. Dafür werden zwei Zeichen gebraucht. Da acht Pixelzeilen einen String ergeben, sind auch hierbei 16 Zeichen für eine Definition notwendig.

CALL LINK ("KREIS", Y,X,R)–->
Dieser Befehl erlaubt es, Kreise mit dem Mittelpunkt X,Y und dem Radius R zu erzeugen. Der Radius darf max. 255 betragen, jedoch kommt es bei einem Radius größer als 185 zu einer Streuung der Pixel, d.h., daß der Kreis nicht mehr geschlossen ist.

CALL LINK ("CKREIS", Y,X,R)–->
Siehe 'KREIS'.
Dieser Befehl löscht die Kreise.

CALL LINK ("LINE", Y1,X1,Y2,X2)–->
Dieser Befehl erlaubt es, Linien von der Koordinate X1,Y1 zu der Koordinate X2,Y2 zu ziehen.

CALL LINK ("CLINE", Y1,X1,Y2,X2)–->
Siehe 'LINE'.
Dieser Befehl löscht die Linien.

CALL LINK ("WLINE", Y,X,Länge, Winkel zur Horizontalen)–->
Dieser Befehl erlaubt es, Linien von der Koordinate X,Y, mit einer maximalen Länge von 320 Pixel und einem Winkel von 0 bis 359 Grad zu ziehen. Der Winkel dreht dabei im Uhrzeigersinn.

CALL LINK ("CWLINE", Y,X,Länge, Winkel zur Horizontalen)–->
Siehe 'WLINE'.
Dieser Befehl löscht die Linien.

CALL LINK ("SAVE", "Dateiname")–->
Dieser Befehl erlaubt es, die mittels CALL LINK ("S

Dieser Befehl bietet die Möglichkeit, eine Grafik auf Diskette abzuspeichern.

CALL LINK ("OLD", "Dateiname")–->
Dieser Befehl erlaubt es, die mittels CALL LINK ("SAVE",...) abgespeicherte Grafik wieder zu laden.

CALL LINK ("HCOPY")–->
Mit diesem Befehl können Hardcopys der Grafik auf einem Seikosha GP 100 A über die PIO der RS 232-Karte angefertigt werden.

CALL LINK ("ZEIG", Sekunden)–->
Dieser Befehl bringt für eine Zeit von maximal 480 Sekunden die Grafik auf den Bildschirm und kehrt danach zum Titelbildschirm zurück.

Abschließend sei noch gesagt, daß die Pixelzeilen 1 bis 192 und die Pixelspalten 1 bis 256 erlaubt sind. Nur die Befehle, wo Zeile und Spalte steht, verlangen die normalen Zeilen- und Spaltenwerte. Auch steht, wie Sie wohl bereits gemerkt haben werden, die Y-Koordinate an erster Stelle und nicht wie mathematisch richtig, zuerst die X-Koordinate. Da wir aber sonst auch immer erst die Zeile und dann die Spalte eingeben, erschien mir diese Anordnung besser.
Und nun wünsche ich Ihnen viel Spaß beim Eintippen.

Bernd Bertling

Anm. d. Red.: Der Bit-Map-Mode ist bekanntlich mit Basic nicht richtig in Einklang zu bringen, deshalb hat der Autor hier die Lösung gewählt, daß das Bild „blind" aufgebaut wird und erst am Schluß für eine bestimmte Zeit gezeigt wird. Das Basic-Programm muß daher immer, bevor CALL LINK ("ZEIG",X) ausgeführt wird, abgespeichert werden, da anschließend zum Titelbild zurückgeführt wird und damit das im Speicher befindliche Basicprogramm verloren ist.

ASSEMBLER IN TI-BASIC

Der Assembler gliedert sich in drei Teile.
Teil -1-, der Editor, steuert alle erforderlichen Eingaben. Der Editor stellt verschiedene Unterprogramme zur Verfügung, um eine Quelldatei einzugeben und zu bearbeiten. Es werden einige Hilfen gegeben, um die Arbeit mit dem Assembler zu erleichtern.
Teil -2-, der eigentliche Assembler, verarbeitet neben den 64 auf der TI-Konsole möglichen Mnemonic und den verschiedenen Adressierungsarten einige Direktiven sowie Pseudo-Instruktionen.
Teil -3- umfaßt verschiedene Unterprogramme, die Umrechnungs- und Berechnungsroutinen, die Fehlerausgabe und Kontrollroutinen enthalten
Das gesamte Programm nimmt 8440 Bytes des Speichers in Anspruch.
Nach dem Start des Programmes stehen im TI-Basic etwas über 4 K-Byte für die Quelldatei zur Verfügung.
Nach einigen Änderungen ist das Programm auch fürs EX-Basic mit Speichererweiterung geeignet. Die Änderungen umfassen insbesondere die Umrechnungsroutinen zwischen Dezimal- und Hexadezimalzahlen.
Ein Listing zeigt die erforderlichen Änderungen.
Die Fähigkeiten des EX-Basic sind aus Gründen der Kompatibilität nicht genutzt.
So kann die Anzahl der Label und der Umfang der Quelldatei verdoppelt werden; eine Straffung des Programmes und die Beschleunigung des Assemblerdurchlaufes sind möglich.

Zur Eingabe:
Grundsätzlich erfolgen alle Eingaben als Stringvariable. Es stehen bei der Eingabe alle Möglichkeiten des TI-Basic zur Verfügung, innerhalb des Strings stehen die Editiermöglichkeiten des TI-Basic zur Wahl.
Durch diese Eingabe müssen einige Besonderheiten beachtet werden.
Eintragungen in die Quelldatei dürfen kein Komma enthalten (ASCII-Zeichen 44).
Abschließende Leerzeichen (ASCII-Zeichen 32) sind nicht möglich, da sie vom TI-Basic gestrichen werden.
Änderungen in der Quelldatei sind nur durch Neueingabe des Datensatzes möglich.
Bei der Eingabe ist das Format des Datensatzes vorgeschrieben. Das Format muß eingehalten werden, da sonst eine Fehlermeldung gegeben ist.
Die Länge eines Datensatzes darf zunächst nicht 40 Zeichen überschreiten.
Die Fehler des Satzes sind genau bestimmt.
Beispiel für das Format:

```
100 LB MOVB
@<834A(R14).R0
12345678901234567890
          1         2
12345678901234567890
          3         4
```

Die Zeichen 4, 7, 12 müssen <Space>sein (ASCII-Zeichen 32).
Zeichen 1 bis 3 enthalten die Zeilennummer.
Zeichen 5 und 6 umfassen das Kennzeichenfeld (für Label)
Zeichen 8 bis 11 stellen das Operationsfeld dar.
Ab Zeichen 13 beginnt das Operandenfeld.
Ein Kommentarfeld ist nicht vorgesehen.

Zeilennummer:
Der Wert der Nummer muß zwischen 100 und 999 liegen und muß ganzzahlig sein. Die Sätze innerhalb der Quelldatei werden an Hand der Zeilennummer sortiert.

Label:
Die Länge des Label ist auf zwei Zeichen begrenzt. Das erste Zeichen muß alphabetisch sein. Das zweite Zeichen ist beliebig. Wird ein Label mit einem Buchstaben, gefolgt von <Space> benutzt, so kann <Space> im Operandenfeld weggelassen werden.
Wird kein Label gewünscht, so sind zwei Leerzeichen einzugeben.
Die Quelldatei darf nur insgesamt 30 Label enthalten.

Operationsfeld:
Hier stehen ein Mnemonic, eine Pseudo-Instruktion oder eine Direktive.
Diese müssen 1 bis 4 Zeichen lang sein.
Eine Kontrolle auf Gültigkeit erfolgt, wie bei den Labeln und den Operanden, erst beim Assemblerdurchlauf.

Operandenfeld:
Es darf kein <Space> innerhalb des Operandenfeldes stehen.
Wird ein Register adressiert, so muß vor der Registerzahl immer ein "R" stehen.
Es sind nur eindeutige Ausdrücke erlaubt.
Konstanten können als Dezimalzahlen (0 – 65535), als negative Dezimalzahlen (–1 bis – 32768) oder als Hexadezimalzahlen (>0000 bis >FFFF) eingegeben werden.
Hexadezimalzahlen haben vorne ein 'größer als' Zeichen (">") und sind 1–4 Zeichen lang. Führende Nullen können weggelassen werden.
Symbole, mit denen auf Labels Bezug genommen wird, dürfen nur zwei Zeichen lang sein. Die Zahl der Symbole ist auf 40 begrenzt.
Der vorliegende Assembler weist zusätzlich die Möglichkeit auf, bestimmte Systemadressen durch Nennen des Namens zu benutzen. Hierbei handelt es sich um wichtige Schreib- und Leseadressen sowie BLWP-Vektoren. Die Adressen werden berechnet, wenn der Systemadressenname mit vorangehendem "at"-Zeichen (-@-) im Operandenfeld steht.

Im Basic stehen 30, im EX-Basic 42 Systemadressen so zur Verfügung (siehe Liste).

Der Vorteil ist, daß nicht die Adresse, sondern der wesentlich einfacher zu merkende Name im Programm steht, wodurch auch die Lesbarkeit eines Programmes erhöht wird.
Beispiele:

```
160 BF BLWP @VSBW
220 MOVB @GRMRD.R0
```

Beim Eingeben eines Datensatzes wird zunächst nur die Zeilennummer auf Gültigkeit, die Einhaltung des Formates sowie das Operationsfeld geprüft.
Ob ein Label, ein Mnemonic oder Operand gültig ist, überprüft erst der Assembler beim assemblieren. Der Assembler gibt eventuelle Fehlermeldungen zur Fehlerroutine. Der Assemblerdurchlauf wird gestoppt, die Fehlermeldungen ausgegeben und dann zur Eingabezeile verzweigt.
Der Fehler muß, um einen erneuten Abbruch zu verhindern, vor dem nächsten Assemblerdurchlauf beseitigt werden.

Neben der Eingabe von Datensätzen können die Unterprogramme des Editors angewählt werden. Dies geschieht durch die Eingabe eines Zeichens gefolgt von <ENTER>. Der Editor stellt folgende Unterprogramme zur Verfügung:

"N":
Die Anzahl der Sätze wird auf -0- gesetzt und der Bildschirm gelöscht

"H":
Der Bildschirm wird gelöscht.

"A":
Verzweigung zum Assembler. Erst wird geprüft, ob die erste Anweisung "AORG" und die letzte "END" ist. Wenn nicht, wird mit Fehlermeldung abgebrochen. Dann werden einige Variablen gesetzt (eventuell vorhandene Label eines vorhergehenden Assemblerdurchlaufes werden u.a. gelöscht) und zum Assembler gesprungen.
Der Assemblerdurchlauf kann durch Drücken der <ENTER>-Taste unterbrochen werden. Das Programm verzweigt dann zurück zur Eingabezeile.

"M":
Zeigt die Anzahl der Datensätze sowie die noch freien an. Nach einem Assemblerdurchlauf werden die benutzten Label und die dazugehörenden Adressen ausgegeben, um eine Verknüpfung zweier Programme zu ermöglichen.
Dieses Unterprogramm wird nach jedem fehlerfreien Assemblerdurchlauf durchgeführt.

"S":
Nach der Verzweigung wird die Eingabe einer Adresse erwartet (Dezimal oder Hexadezimal). Ab dieser Adresse werden die Speicherinhalte (Worte) in hexadezimaler Schreibweise ausgegeben. Wird die <Space>-Taste gedrückt, springt das Programm zur Eingabezeile zurück.

"L":
Dieses Programm listet aus der Quelldatei. Folgende Eingaben sind nach Verzweigung möglich:
"A"-<ENTER>: Listet ab der Zeilennummer. Es werden zunächst 21 Sätze ausgegeben. Wird das Ende der Quelldatei erreicht, springt die Kontrolle zur Eingabezeile zurück.
Ist das Ende nicht erreicht, wird ein weiteres Kommande erwartet:
Taste -<ENTER>- springt zur Eingabezeile zurück.
Taste -<Space>- listet nur den nächsten Datensatz.
Jede andere Taste führt dazu, daß wiederum 21 Sätze ausgegeben werden.

"R":
Mit dieser Unterroutine kann ein Maschinenspracheprogramm gestartet werden. Nach der Verzweigung wird die Eingabe des Programmnamens erwartet. Dieser muß in der DEF-Table des entsprechenden Modules (Minimem bzw. Erweiterung) stehen. Sollen Parameter übergeben werden, so muß die entsprechende Basic-Zeile (1230) geändert werden.
Nach <ENTER> wird zum Maschinensprache-Programm verzweigt (mittels "LINK").
Steht der Programmname nicht in der DEF-Table kommt es zu einer Basic-Fehlermeldung mit Programmabbruch des Assemblers!

" ":
Nach der Eingabe einer Dezimalzahl (0–65535) wird die Zahl in hexadezimaler Schreibweise ausgegeben. Dezimalzahlen über 32767 werden zusätzlich als "Zweier-Komplement" ausgegeben.

"-":
Nach der Eingabe einer negativen Dezimalzahl (–1 bis –32768) wird die Zahl in hexadezimaler Schreibweise sowie als Dezimalzahl (über 32767) ausgegeben.

" ":
Nach der Eingabe einer hexadezimalen Zahl (1 bis 4 Zeichen lang) wird diese ins Dezimalsystem umgerechnet und ausgegeben. Zahlen über 32767 werden zusätzlich als "Zweier-Komplement" ausgegeben.
Bei der Ausgabe der negativen Dezimalzahlen kommt es beim EX-Basic zu Fehlern, da, anders als beim Minimem, die Dezimalzahlen auf -null- runtergerechnet werden. Die Funktionen erschienen nicht wichtig genug, um die Änderungen im Listing aufzunehmen. Mittels Zwischenvariablen kann dies einfach geändert werden.

"C":
Verzweigt zu den Cassettenrekorderroutinen.
Es kann gewählt werden, ob eine Quelldatei eingespielt wird oder auf Cassette gesichert wird.
Vor jedem Start eines neuen Maschinensprache-Programmes sollte die Quelldatei auf Cassette überspielt werden.
„Hängt" sich nämlich das Programm auf, so muß fast immer der Computer ausgeschaltet werden. In diesem Fall würde die erstellte Quelldatei verloren gehen.
Trotz der langsamen Cassettenroutine ist dieses Abspeichern immer noch schneller, als die Neueingabe über die Tastatur.

"Z":
Dieses Unterprogramm stattet die Datensätze mit neuen Zeilennummern aus.
Die Zeilen beginnen bei 100 und steigen je Zeile um 5 (wie RES 100,5 des TI-Basic).
Andere Zeichen als die o.a. führen zu Fehlermeldungen. Sollen Datensätze aus der Quelldatei gelöscht werden, so muß lediglich die Zeilennummer eingegeben werden. Änderungen der Sätze sind nur durch Neueingabe möglich. Wird die Funktion "N" irrtümlich angewählt, so sind die eingegebenen Sätze nicht verloren. Im Direktmodus muß dann nach "Break" in die Variable -S- die Zahl der Sätze eingegeben werden. Mit CON wird der Assembler dann wieder gestartet.
Folgende Pseudo-Instruktionen bzw. Direktiven werden vom Assembler interpretiert, wenn sie im Operationsfeld stehen:

Allgemeines:
Bei der Beschreibung wird das Format an Hand einer Beispielszeile gegeben. Die ersten drei Zeichen stellen immer die Zeilennummer dar. Steht ein Label im Kennzeichenfeld, so ist die Benutzung von diesen wahlfrei (auf Besonderheiten wird hingewiesen).
a = Beispiel
b = Beschreibung

1. "NOP"
a) 100 LB NOP
b) Pseudoinstruktion; identisch mit JMP +2. Das Operandenfeld wird völlig ignoriert.

2. "RT"
a) 520 B1 RT
b) Pseudoinstruktion; identisch mit B *R11 Das Operandenfeld wird völlig ignoriert.

3. "DEF"
a) 670 LB DEF WORT. @Operand
b) WORT kann 1 – 6 Zeichen lang sein (wird immer intern mit Leerzeichen (ASCII 32) auf 6 Zeichen aufgefüllt). WORT wird als Programmname eines Maschinensprache-Programms in die DEF-Table eingetragen. Der Operand nach dem "at"-Zeichen kann eine Hexadezimal- oder eine Dezimalzahl oder ein Symbol sein. Es muß die Adresse enthalten, an der WORT startet. Die Adresse wird in die DEF-Table eingetragen. Das Unterprogramm "DEF" verändert Zeiger auf die DEF-Table im RAM. Treten nach "DEF" Fehler im Assemblerprogramm auf, so springt der Assembler zur Eingabezeile zurück. Vor einem neuen

Assemblerdurchlauf muß der Zeiger im Direktmodus zurückgesetzt werden. Es ist daher sinnvoll, "DEF" erst unmittelbar vor der "END "-Zeile zu benutzen. Trotzdem sollte der Wert erst gesichert werden. Beim Minimem steht der Wert an Adresse >701E (28702), beim EX-Basic in >2004 (8196). Mit dem Unterprogramm "S" ist der Wert leicht festzustellen.
Wird ein Label benutzt, so wird dieser der momentanen Adresse zugewiesen.
Im Regelfall sollte kein Label benutzt werden.

4. "EQU " nn

a) 130 WT EQU >7D00
b) Einem Label wird ein bestimmter Wert (nn) zugewiesen, der zwischen 0 und 65535 liegen muß. Ein Label muß benutzt werden, da sonst ein Fehler auftritt.

5. "BSS " nn

a) 130 BF BSS 32
b) Es werden nn Bytes Speicherplatz reserviert. Sie haben keinen vordefinierten Wert.
Ist der Wert nn ungerade, so wird ein Byte dazuaddiert, damit die nächste Anweisung an einer geraden Adresse beginnt.
Ein Label muß benutzt werden, da sonst ein Fehler auftritt. Die Adresse des ersten Byte wird dem Label zugeordnet.
Diese Direktive wird benutzt, um Stringbuffer oder Platz für ein Workspace zu reservieren.

6. "DATA" nn

a) 126 VW DATA >2000
b) Die Direktive weist dem nächsten Wort den Wert nn zu. Der Wert muß zwischen 0 und 65535 liegen. Es können auch Symbole als Operand benutzt werden.
Es ist immer nur ein Wert pro DATA-Direktive möglich.

7. "TEXT"

a) 980 AS TEXT ASCII-Zeichen
b) Diese Direktive speichert ab der augenblicklichen Adresse einen Text ab. Alle ASCII-Zeichen, die über die Tastatur eingegeben werden können sind erlaubt.
Der Text beginnt ab dem 13. Zeichen des Datensatzes. Er muß nicht in Anführungsstriche gesetzt werden.
28 Zeichen können pro Text-Direktive eingegeben werden.
Komma wird als FCTN -C- (ASCII-Zeichen 96) eingegeben.
Abschließende Leerzeichen sind nicht möglich, da diese bei der Eingabe vom TI-Basic gestrichen werden.
Die beiden Einschränkungen gelten nicht, wenn der gesamte Datensatz in Anführungsstrichen eingegeben wird.
Auf das Format ist hierbei zu achten. Es dürfen nie 5 TEXT-Direktiven mit 28 Zeichen Text hintereinander in der Quelldatei stehen, da dies die Cassettenroutinen nicht erlauben. Es käme zu einer TI-Basic-Fehlermeldung mit Abbruch des Assemblers.
Ein Programm darf auch nicht viele TEXT-Direktiven mit längerem Text beinhalten. Es könnte zu einer TI-Basic-Fehlermeldung (Memory Full) führen.
Wird die Anzahl der Zeichen im Operandenfeld auf 16 beschränkt, so treten beide Möglichkeiten nicht auf. Zudem bleibt das Listing der Quelldatei übersichtlich.
Wird eine ungerade Anzahl von Text-Zeichen 32 <Space> an, damit die nächste Anweisung an einer geraden Adresse beginnt.
Die Adresse, an der das erste Zeichen steht, wird einem benutzten Label zugewiesen.

8. " AORG"

a) 100 AORG >7118
b) Die AORG-Anweisung muß immer im Operationsfeld des ersten Datensatzes stehen.
Ein benutzter Label wird ignoriert.
Die Benutzung des Operanden ist wahlfrei. Wird kein Operand benutzt, so liest die Anweisung die nächste freie Adresse (FFAM) aus dem Speicher und ab dieser Adresse wird das Maschinensprache-Programm ins RAM abgespeichert.
Die Werte werden beim Minimem aus der Adresse >701C (28700 dez.) und aus dem EX-Basic aus >2002 (8294 dez.) ausgelesen.
Wird ein Operand benutzt, so wird dieser berechnet. Ab dieser Adresse erfolgt die Abspeicherung.
Symbole sind nicht erlaubt.
Der Assembler führt nur eine geringe Kontrolle durch, ob die selbstgewählte Adresse sinnvoll ist (ob vorhandene Programme überschrieben werden, dies wird nicht überprüft).
Beim Minimem ist der Bereich von >7000 – >7FFF, beim EX-Basic der von> 2474 – >3FFF zulässig.
Werden die Basic-Zeilen 680 und 1930 des Assemblers gestrichen, so ist jeder Speicherbereich zulässig.

9. "END "

a) 999 END #
b) Die "END "-Direktive muß immer im letzten Datensatz der Quelldatei stehen.
Wird ein Label benutzt, so wird dieser ignoriert.

Folgende Adressen sind durch Erwähnung des Namens mit vorangehenden "at"-Zeichens erreichbar:

Name:	Adresse (hexadezimal) a) Mini-Memory	b) EX-Basic
UTLTAB	7020	----
PAD	8300	8300
GPLWS	83E0	83E0
SOUND	8400	8400
VDPRD	8800	8800
VDPSTA	8802	8802
VDPWD	8C00	8C00
VDPWA	8C02	8C02
SPCHRD	9000	9000
SPCHWT	9400	9400
GRMRD	9800	9800
GRMRA	9802	9802
GRMWD	9C00	9C00
GRMWA	9C02	9C02
SCAN	000E	000E
XMLINK	601C	2018
KSCAN	6020	201C
VSBW	6024	2020
VMBW	6028	2024
VSBR	602C	2028
VMBR	6030	202C
VWTR	6034	2030
DSRLNK	6038	----
LOADER	603C	----
GPLLNK	6018	----
NUMASG	6040	2008
NUMREF	6044	200C
STRASG	6048	2010
STRREF	604C	2014
ERR	6050	2034
FAC	----	834A
FADD	----	0D80
FSUB	----	0D7C
FMUL	----	0E88
FDIV	----	0FF4
SADD	----	0D84
SSUB	----	0D74
SMUL	----	0E8C
SDIV	----	0FF8
CSN	----	11AE
CFI	----	12B8
FCOMP	----	0D3A
NEXT	----	0070
ASCII	----	8375
STATUS	----	837C
ARG	----	835C

Änderungen beim Minimem sind nicht möglich, da hier die Adressen aus Platzgründen aus dem ROM ausgelesen werden.
Änderungen beim EX-Basic sind möglich, da Namen und dazugehörende Adressen im Basic-Programm definiert werden.
Beispiele:

```
100 lb BLWP @XMLINK
110    MOV @FAC.
       @ARG
120 A MOVB @ASCII.
       R1
130    CB  @STATUS.
       R7
```

```
100 REM ASSEMBLER
110 REM (c) UWE WIEMER '85
120 DIM S$(150),O(3),L(30),E
    (40),E$(40)
130 D$="XOP  NOP RT   DEF EQU
    BSS DATATEXT"
140 M$="A   AB  ABS AI  ANDI
    B   BL  BLWPC   CB  CI  CLR
    COC CZC DEC DECTDIV INC INCT
    INV JEQ JGT JH  JHE JL  JLE"
150 M$=M$+"  JLT JMP JNC JNE
    JNO JOC JOP LDCRLI  LIMILWPI
    MOV MOVBMPY NEG ORI RTWPS
    SB  SBO SBZ SETOSLA SOC "
160 M$=M$+"SOCBSRA SRC SRL S
    TCRSTSTSTWPSWPBSZC SZCBTB  X
        XOR "
170 C$="A001B001074602280248
    04460686044068001900102880
    2003240306060646306305860
    0546130215021B021402102"
180 C$=C$+"12021102100217021
    6021902180210230004020803080
    2E8C001D001380305060268038
    0017001101D021E020706A05E001"
190 C$=C$+"F001080508050950
    4040206802A806C64001500011F020
    4862803"
200 S=0
210 CALL CLEAR
220 PRINT :"#___ __ #___ OP"
230 INPUT "":T$
240 D=LEN(T$)
250 IF (D>37)+(D=0)THEN 560
260 ON POS("NHAMSLR>.-CZ",T$
    ,1)+1 GOTO 270,200,210,580,7
    30,810,970,1210,1250,1310,13
    70,1420,1590
270 V$=SEG$(T$,1,3)
280 GOSUB 3960
290 IF ZW<100 THEN 3560
300 IF D=3 THEN 380
310 IF S=150 THEN 3730
320 T$=T$&"   "
330 IF (SEG$(T$,4,1)<>" ")+(
    SEG$(T$,7,1)<>" ")+(SEG$(T$,
    8,1)=" ")+(SEG$(T$,12,1)<>"
    ")THEN 560
340 IF D>12 THEN 360
350 D=11
360 T$=SEG$(T$,1,D)
370 IF (V$>SEG$(S$(S),1,3))+
    (S=0)THEN 530
380 GOSUB 4020
390 IF (D=3)+(V$<>SEG$(S$(I)
    ,1,3))=-2 THEN 550
400 IF V$<SEG$(S$(I),1,3)THE
    N 470
410 IF D<>3 THEN 510
420 FOR J=I TO S-1
430 S$(J)=S$(J+1)
440 NEXT J
450 S=S-1
460 GOTO 550
470 FOR J=S TO I STEP -1
480 S$(J+1)=S$(J)
490 NEXT J
500 S=S+1
510 S$(I)=T$
520 GOTO 550
530 S=S+1
540 S$(S)=T$
550 GOTO 230
560 PRINT :"EINGABE?"
570 GOTO 4290
580 T$=S$(1)
590 IF (SEG$(S$(S),8,4)<>"EN
    D ")+(SEG$(T$,8,4)<>"AORG")T
    HEN 4270
600 IF LEN(T$)<12 THEN 650
610 O$=SEG$(T$,13,6)
620 GOSUB 4070
630 A=D-(D/2<>INT(D/2))
640 GOTO 670
650 CALL PEEK(28700,I,J)
660 A=I*256+J
670 PRINT : : :"AORG";A; : :
680 IF A<28672 THEN 950
690 M=1
700 Z=1
710 L$=""
720 GOTO 1630
730 PRINT :"SAETZE:";S,"  ="
    ;150-S;"FREI"
740 PRINT "LABEL :";Z-1-(Z=0
    ); :
750 FOR I=1 TO Z-1
760 IF I>9 THEN 780
770 PRINT " ";
780 PRINT ;STR$(I);". ";SEG$
    (L$,I*2+1,2);" @";STR$(L(I))
    ,
790 NEXT I
800 GOTO 220
810 INPUT "ADR.:":O$
820 GOSUB 4070
830 K=D
840 IF (K<-32768)+(K>65536)T
    HEN 950
850 IF K<32767 THEN 870
860 K=K-65536
870 CALL PEEK(K,D,J)
880 D=D*256+J
890 GOSUB 3840
900 PRINT K;TAB(10);"=> ";H$
910 K=K+2
920 CALL KEY(0,R,T)
930 IF R=13 THEN 220
```

```
100 REM ASSEMBLER
110 REM (c) UWE WIEMER '85
120 DIM S$(150),O(3),L(30),E
    (40),E$(40)
130 D$="XOP  NOP RT   DEF EQU
    BSS DATATEXT"
140 M$="A   AB  ABS AI  ANDI
    B   BL  BLWPC   CB  CI  CLR
    COC CZC DEC DECTDIV INC INCT
    INV JEQ JGT JH  JHE JL  JLE"
150 M$=M$+"  JLT JMP JNC JNE
    JNO JOC JOP LDCRLI  LIMILWPI
    MOV MOVBMPY NEG ORI RTWPS
    SB  SBO SBZ SETOSLA SOC "
160 M$=M$+"SOCBSRA SRC SRL S
    TCRSTSTSTWPSWPBSZC SZCBTB  X
        XOR "
170 C$="A001B001074602280248
    04460686044068001900102880
    2003240306060646306305860
    0546130215021B021402102"
180 C$=C$+"12021102100217021
    6021902180210230004020803080
    2E8C001D001380305060268038
    0017001101D021E020706A05E001"
190 C$=C$+"F001080508050950
    4040206802A806C64001500011F020
    4862803"
200 S=0
210 CALL CLEAR
220 PRINT :"#___ __ #___ OP"
230 INPUT "":T$
240 D=LEN(T$)
250 IF (D>37)+(D=0)THEN 560
260 ON POS("NHAMSLR>.-CZ",T$
    ,1)+1 GOTO 270,200,210,580,7
    30,810,970,1210,1250,1310,13
    70,1420,1590
270 V$=SEG$(T$,1,3)
280 GOSUB 3960
290 IF ZW<100 THEN 3560
300 IF D=3 THEN 380
310 IF S=150 THEN 3730
320 T$=T$&"   "
330 IF (SEG$(T$,4,1)<>" ")+(
    SEG$(T$,7,1)<>" ")+(SEG$(T$,
    8,1)=" ")+(SEG$(T$,12,1)<>"
    ")THEN 560
340 IF D>12 THEN 360
350 D=11
360 T$=SEG$(T$,1,D)
370 IF (V$>SEG$(S$(S),1,3))+
    (S=0)THEN 530
380 GOSUB 4020
390 IF (D=3)+(V$<>SEG$(S$(I)
    ,1,3))=-2 THEN 550
400 IF V$<SEG$(S$(I),1,3)THE
    N 470
410 IF D<>3 THEN 510
420 FOR J=I TO S-1
430 S$(J)=S$(J+1)
440 NEXT J
450 S=S-1
460 GOTO 550
470 FOR J=S TO I STEP -1
480 S$(J+1)=S$(J)
490 NEXT J
500 S=S+1
510 S$(I)=T$
520 GOTO 550
530 S=S+1
540 S$(S)=T$
550 GOTO 230
560 PRINT :"EINGABE?"
570 GOTO 4290
580 T$=S$(1)
590 IF (SEG$(S$(S),8,4)<>"EN
    D ")+(SEG$(T$,8,4)<>"AORG")T
    HEN 4270
600 IF LEN(T$)<12 THEN 650
610 O$=SEG$(T$,13,6)
620 GOSUB 4070
630 A=D-(D/2<>INT(D/2))
640 GOTO 670
650 CALL PEEK(28700,I,J)
660 A=I*256+J
670 PRINT : : :"AORG";A; : :
680 IF A<28672 THEN 950
690 M=1
700 Z=1
710 L$=""
720 GOTO 1630
730 PRINT :"SAETZE:";S,"  ="
    ;150-S;"FREI"
740 PRINT "LABEL :";Z-1-(Z=0
    ); :
750 FOR I=1 TO Z-1
760 IF I>9 THEN 780
770 PRINT " ";
780 PRINT ;STR$(I);". ";SEG$
    (L$,I*2+1,2);" @";STR$(L(I))
    ,
790 NEXT I
800 GOTO 220
810 INPUT "ADR.:":O$
820 GOSUB 4070
830 K=D
840 IF (K<-32768)+(K>65536)T
    HEN 950
850 IF K<32767 THEN 870
860 K=K-65536
870 CALL PEEK(K,D,J)
880 D=D*256+J
890 GOSUB 3840
900 PRINT K;TAB(10);"=> ";H$
910 K=K+2
920 CALL KEY(0,R,T)
930 IF R=13 THEN 220
```

FEHLERMELDUNGEN:

Die Fehlermeldungen sind in der Regel eindeutig, da zu dem Zeitpunkt, zu dem sie ausgegeben werden, auch der Fehler auf dem Bildschirm zu sehen ist.
In der Regel befindet sich der Fehler in der letzten ausgegebenen Zeile.

Einige Besonderheiten:

"Überlauf" –
während der Eingabe der Datensätze: es sind mehr als 150 Sätze vorhanden
– während des Assembliervorganges: es sind mehr als 30 Label oder 40 Symbole vorhanden

"?" mit Datensatz, Adresse und Label
– es wird ein Symbol benutzt und es gibt keinen entsprechenden Label
– das letzte Zeichen eines Labels und das erste Zeichen des nächsten Labels ergeben zusammen ein benutztes Symbol

"Eingabe"
– wird ein Datensatz eingegeben, stimmt das Format nicht bzw. das Operationsfeld ist leer
– unzulässige Eingaben im Editor

"Operand/Znr?"
– wird ein Datensatz eingegeben, so stimmt die Zeilennummer nicht
– im Editor: falsche Dezimal- oder Hexadezimalzahl
– während des Assemblerdurchlaufes: im Operandenfeld steht ein ungültiger Operand

"Label"
– das erste Zeichen eines Labels ist nicht alphabetisch
tritt dies bei einem Symbol auf, so wird die Meldung "Operand/Znr?" gegeben.

"Adr.?"
– es wird an eine unzulässige Adresse geladen
– falsche Adresse (>65535 oder< -32768)

Die END-Anweisung berechnet die erste freie Adresse im RAM. Dieser Wert wird dem Label "EX" zugewiesen. Auf diese Adresse kann im Programm im Operandenfeld durch das Symbol ("EX") bezug genommen werden.
Dieser Label "EX" sollte in der Quelldatei als Label nicht noch einmal verwandt werden, da es mit Sicherheit zu Fehlberechnungen kommt und ein Maschinensprache-Programm mit Sicherheit abstürzt.
Die Benutzung des "File"-Zeichens ist wahlfrei. Wird es benutzt, so wird nach fehlerfreiem Assemblerdurchlauf der Zeiger FFAM (First Free Adress in Memory) nicht geändert.
Dies ist zum Austesten des Assemblers gedacht. Es ist aber auch wichtig, wenn einzelne Worte eines Maschinensprache-Programms geändert werden sollen.
Beispiel:
100 AORG Adresse
110 DATA WERT
120 END #
Auf FFAM wird nicht eingewirkt.
Wird das "File"-Zeichen weggelassen, so wird FFAM geändert (gleich zu "EX" gesetzt).

10." ! "

a) 120 ! Kommentar
b) Benutzte Label werden ignoriert.
Da kein Kommentarfeld in den Datensätzen vorgesehen ist, wurde die Möglichkeit eingeräumt, Kommentare in eigenen Datensätzen ins Assemblerlisting einzufügen.
Das Program wird lesbarer.
Es gelten die selben Einschränkungen hinsichtlich der Länge wie bei der Anweisung "TEXT".

Uwe Wiemer

```
940 IF T<>0 THEN 920 ELSE 85
0
950 PRINT :"ADR.?"
960 GOTO 4290
970 INPUT "ZNR.: ":T$
980 IF (T$="")+(LEN(T$)>3)TH
EN 560
990 IF T$="A" THEN 1080
1000 IF T$<>"L" THEN 1040
1010 IF S<14 THEN 1080
1020 D=S-14
1030 GOTO 1080
1040 V$=SEG$(T$,1,3)
1050 GOSUB 3970
1060 GOSUB 4020
1070 D=I
1080 K=0
1090 FOR I=D TO S
1100 K=K+1
1110 PRINT S$(I)
1120 IF K<21 THEN 1190
1130 CALL KEY(0,R,T)
1140 IF T=0 THEN 1130
1150 IF R=13 THEN 220
1160 K=20
1170 IF R=32 THEN 1190
1180 K=0
1190 NEXT I
1200 GOTO 230
1210 INPUT "NAME: ":Q$
1220 IF Q$="" THEN 560
1230 CALL LINK(Q$)
1240 GOTO 220
1250 INPUT "HEX:>":H$
1260 GOSUB 3890
1270 PRINT D
1280 IF D<32768 THEN 1300
1290 PRINT D-65536
1300 GOTO 220
1310 INPUT "DEZ: ":V$
1320 GOSUB 3960
1330 D=ZW
1340 GOSUB 3830
1350 PRINT ">";H$
1360 GOTO 1280
1370 INPUT "DEZ: -":V$
1380 GOSUB 4100
1390 GOSUB 3830
1400 PRINT D:" >";H$
1410 GOTO 220
1420 INPUT "1=SAVE 2=OLD (CS
1): ";V$
1430 IF V$="2" THEN 1520
1440 IF V$<>"1" THEN 560
1450 OPEN #1:"CS1",INTERNAL,
OUTPUT,FIXED 192
1460 FOR I=1 TO 150 STEP 5
1470 PRINT #1:S,S$(I),S$(I+1
),S$(I+2),S$(I+3),S$(I+4)
1480 IF S$(I+4)="" THEN 1500
1490 NEXT I
1500 CLOSE #1
1510 GOTO 1580
1520 OPEN #1:"CS1",INTERNAL,
INPUT ,FIXED 192
1530 FOR I=1 TO 150 STEP 5
1540 INPUT #1:S,S$(I),S$(I+1
),S$(I+2),S$(I+3),S$(I+4)
1550 IF S$(I+4)="" THEN 1570
1560 NEXT I
1570 CLOSE #1
1580 GOTO 220
1590 FOR I=1 TO S
1600 S$(I)=STR$(95+I*5)&SEG$
(S$(I),4,34)
1610 NEXT I
1620 GOTO 220
1630 FOR J=2 TO S-1
1640 T$=S$(J)
1650 PRINT T$
1660 IF SEG$(T$,8,1)="!" THE
N 1980
1670 CALL KEY(0,R,T)
1680 IF R=13 THEN 220
1690 B=0
1700 IF SEG$(T$,5,2)="  " TH
EN 1790
1710 IF (SEG$(T$,5,1)>="A")+
(SEG$(T$,5,1)<="Z")=-2 THEN
1740
1720 PRINT :"LABEL?"
1730 GOTO 4290
1740 IF Z>30 THEN 3730
1750 L$=L$&SEG$(T$,5,2)
1760 L(Z)=A
1770 B=1
1780 Z=Z+1
1790 W=1
1800 B$=SEG$(T$,8,4)
1810 P=POS(M$,B$,1)
1820 IF P THEN 1870
1830 ON INT((POS(D$,B$,1)+3)
/4)+1 GOTO 4280,1840,2730,27
50,2770,2940,2990,3050,3090
1840 K=9
1850 C=11264
1860 GOTO 1910
1870 CALL CHAR(143,SEG$(C$,P
,3))
1880 CALL PEEKV(1912,Y,H)
1890 C=Y*256+H
1900 K=VAL(SEG$(C$,P+3,1))
1910 ON K GOSUB 2230,2290,23
90,2390,2470,2550,2330,2590,
2390
1920 FOR I=1 TO W
1930 IF A>32768 THEN 950
1940 R=INT(O(I)/256)
1950 CALL LOAD(A,R,O(I)-R*25
6)
1960 A=A+2
1970 NEXT I
1980 NEXT J
1990 PRINT S$(J):
2000 L$="EX"&L$
2010 L(0)=A
2020 FOR J=1 TO M-1
2030 D=(POS(L$,SEG$(E$(J),1,
2),1)+1)/2
2040 IF D<>INT(D)THEN 2090
2050 D=L(D-1)
2060 IF E(J)>0 THEN 2130
2070 K=(D+E(J)-2)/2
2080 IF (K>127)+(K<-128)=0 T
HEN 2110
2090 PRINT :"?":S$(VAL(SEG$(
E$(J),3,3))):"ADR.:";ABS(E(J
));"LABEL: ";SEG$(E$(J),1,2)
2100 GOTO 4290
2110 CALL LOAD(-E(J)+1,K)
2120 GOTO 2150
2130 I=INT(D/256)
2140 CALL LOAD(E(J),I,D-256*
I)
2150 NEXT J
2160 PRINT :"ENDADR:";A;
2170 IF SEG$(S$(S),13,1)="#"
 THEN 2210
2180 T=INT(A/256)
2190 CALL LOAD(28700,T,A-256
*T)
2200 PRINT ;"  = FFAM!";
2210 PRINT ;"(EX)";
2220 GOTO 730
2230 GOSUB 3240
2240 O(1)=C+16*T+R
2250 O$=SEG$(T$,G+1,12)
2260 GOSUB 3270
2270 O(1)=O(1)+1024*T+64*R
2280 RETURN
2290 O$=SEG$(T$,13,3)&" "
2300 IF SEG$(O$,1,1)<"A" THE
N 2350
2310 GOSUB 4230
2320 E(M-1)=-E(M-1)
2330 D=0
2340 GOTO 2370
2350 GOSUB 4150
2360 IF D>255 THEN 3560
2370 O(W)=C+D
2380 RETURN
2390 GOSUB 3240
2400 O(1)=C+16*T+R
2410 V$=SEG$(T$,G+1-(K=3),2)
2420 IF (K=3)+(SEG$(T$,G+1,1
)<>"R")=-2 THEN 3560
2430 GOSUB 3960
2440 IF ZW>15 THEN 3560
2450 O(1)=O(1)+64*ZW
2460 RETURN
2470 IF SEG$(T$,13,1)<>"R" T
HEN 3560
2480 GOSUB 3240
2490 O(1)=C+R
2500 V$=SEG$(T$,G+1,2)
2510 GOSUB 3960
2520 IF ZW>15 THEN 3560
2530 O(1)=O(1)+16*ZW
2540 RETURN
2550 G=25
2560 GOSUB 3260
2570 O(1)=C+16*T+R
2580 RETURN
2590 IF POS("LIMILWPI",B$,1)
=0 THEN 2630
2600 O(1)=C
2610 G=12
2620 GOTO 2670
2630 IF SEG$(T$,13,1)<>"R" T
HEN 3560
2640 GOSUB 3240
2650 O(1)=C+R
2660 IF SEG$(T$,8,1)="S" THE
N 2720
2670 O$=SEG$(T$,G+1,6)
2680 W=W+1
2690 GOSUB 4070
2700 IF (B$="LIMI")+(D>2)=-2
 THEN 3560
2710 O(2)=D
2720 RETURN
2730 O(1)=4096
2740 GOTO 1920
2750 O(1)=1115
2760 GOTO 1920
2770 G=POS(T$,".",14)
2780 IF (SEG$(T$,G+1,1)<>"@"
)THEN 3560
2790 O$=SEG$(T$,G+2,6)
2800 Q$=SEG$(T$,13,G-13)&"
   "
2810 Q$=SEG$(Q$,1,6)
2820 CALL PEEK(28702,R,T)
2830 DE=A
2840 A=(R*256+T)-8
2850 R=6
2860 GOSUB 3150
2870 GOSUB 4070
2880 R=INT(D/256)
2890 CALL LOAD(A,R,D-R*256)
2900 Z1=INT((A-6)/256)
2910 CALL LOAD(28702,Z1,A-6-
256*Z1)
2920 A=DE
2930 GOTO 1980
2940 IF B=0 THEN 1720
```

```
2950 O$=SEG$(T$,13,6)
2960 GOSUB 4070
2970 L(Z-1)=D
2980 GOTO 1980
2990 IF B=0 THEN 1720
3000 V$=SEG$(T$,13,5)
3010 GOSUB 3960
3020 L(Z-1)=A
3030 A=A+ZW-(INT(ZW/2)<>ZW/2
)
3040 GOTO 1980
3050 O$=SEG$(T$,13,6)
3060 GOSUB 4070
3070 O(1)=D
3080 GOTO 1920
3090 K=LEN(T$)
3100 IF K<13 THEN 3560
3110 Q$=SEG$(T$,13,K-12)
3120 R=LEN(Q$)
3130 R=R-(INT(R/2)<>R/2)
3140 Q$=Q$&" "
3150 FOR I=1 TO R
3160 T=ASC(SEG$(Q$,I,1))
3170 IF T<>96 THEN 3190
3180 T=44
3190 CALL LOAD(A,T)
3200 A=A+1
3210 NEXT I
3220 IF B$="TEXT" THEN 1980
3230 RETURN
3240 G=POS(T$,".",13)
3250 G=G-16*(G=0)
3260 O$=SEG$(T$,13,G-13)
3270 Z$=SEG$(O$,1,1)
3280 ON POS("R*@",Z$,1)+1 GO
TO 3560,3290,3330,3420
3290 T=0
3300 V$=SEG$(O$,2,2)
3310 GOSUB 3960
3320 GOTO 3390
3330 T=1
3340 R=POS(O$,"+",4)
3350 IF R=0 THEN 3370
3360 T=3
3370 V$=SEG$(O$,3,LEN(O$)+(R
<>0)-2)
3380 GOSUB 3960
3390 R=ZW
3400 IF R>15 THEN 3560
3410 GOTO 3820
3420 W=W+1
3430 R=0
3440 T=2
3450 GOTO 3580
3460 Q$=SEG$(O$,2,6)
3470 FOR I=28430 TO 28662 ST
EP 8
3480 FOR R1=0 TO 5
3490 CALL PEEK(I+R1,R2)
3500 IF CHR$(R2)<>SEG$(Q$,R1
+1,1)THEN 3550
3510 NEXT R1
3520 CALL PEEK(I+6,R1,R2)
3530 D=R1*256+R2
3540 GOTO 3810
3550 NEXT I
3560 PRINT :"OPERAND/ZNR?"
3570 GOTO 4290
3580 U=POS(O$,"(",3)
3590 IF U=0 THEN 3660
3600 V=POS(O$,")",6)
3610 IF V=0 THEN 3560
3620 V$=SEG$(O$,U+2,V-U-2)
3630 GOSUB 3960
3640 R=ZW
3650 IF R>15 THEN 3560
3660 U=U-(U=0)*8
3670 IF SEG$(O$,2,1)<"A" THE
N 3790
3680 O$=O$&"    "
3690 IF LEN(O$)>6 THEN 3460
3700 A$=SEG$(O$,2,2)
3710 IF SEG$(A$,1,1)>"Z" THE
N 3560
3720 IF M<40 THEN 3750
3730 PRINT :"UEBERLAUF"
3740 GOTO 4290
3750 E(M)=A+W*2-2
3760 E$(M)=A$&STR$(J)
3770 M=M+1
3780 RETURN
3790 O$=SEG$(O$,2,U-2)
3800 GOSUB 4070
3810 O(W)=D
3820 RETURN
3830 IF D>65536 THEN 3560
3840 Y=INT(D/256)
3850 CALL POKEV(1912,Y,D-Y*2
56)
3860 CALL CHARPAT(143,H$)
3870 H$=SEG$(H$,1,4)
3880 RETURN
3890 FOR I=1 TO LEN(H$)
3900 IF POS("0123456789ABCDE
F",SEG$(H$,I,1),1)=0 THEN 35
60
3910 NEXT I
3920 CALL CHAR(143,H$)
3930 CALL PEEKV(1912,Y,H)
3940 D=(Y*256+H)/2^(16-(4*LE
N(H$)))
3950 RETURN
3960 IF V$="" THEN 3560
3970 FOR I=1 TO LEN(V$)
3980 IF POS("0123456789",SEG
$(V$,I,1),1)=0 THEN 3560
3990 NEXT I
4000 ZW=VAL(V$)
4010 RETURN
4020 FOR I=1 TO S
4030 IF V$<=SEG$(S$(I),1,3)T
HEN 4060
4040 NEXT I
4050 I=I-1
4060 RETURN
4070 IF SEG$(O$,1,1)=">" THE
N 4200
4080 IF SEG$(O$,1,1)<>"-" TH
EN 4140
4090 V$=SEG$(O$,2,6)
4100 GOSUB 3960
4110 IF ZW>32768 THEN 3560
4120 D=65536-ZW
4130 GOTO 4260
4140 IF SEG$(O$,1,1)>="A" TH
EN 4230
4150 V$=O$
4160 GOSUB 3960
4170 D=ZW
4180 IF D>65535 THEN 3560
4190 GOTO 4260
4200 H$=SEG$(O$,2,4)
4210 GOSUB 3890
4220 GOTO 4260
4230 A$=SEG$(O$,1,2)
4240 IF (B$="EQU ")+(T$=S$(1
))+(SEG$(T$,1,1)="S")+(B$="L
IMI")THEN 3560
4250 GOSUB 3710
4260 RETURN
4270 PRINT :"AORG/END";
4280 PRINT :"MNEMONIC ?"
4290 CALL SOUND(190,219,0)
4300 GOTO 220
60 RF$="VDPWA VDPWD VDPRD VD
PSTAFAC   GPLWS PAD   SOUND
SPCHRDSPCHWTGRMRD GRMRA GRMW
D GRMWA SCAN  NUMASGNUMREFST
RASGSTRREFXMLLNKKSCAN "
70 RF$=RF$&"VSBW  VMBW  VSBR
  VMBR  VWTR  ERR   FADD  FS
UB  FMUL  FDIV  SADD  SSUB
SMUL  SDIV  CSN   CFI   FCOM
P NEXT  ASCII STATUSARG   "
80 RFA$="358423584034816348
1
8336103376033536337923686437
8883891238914399363993 8
 8200 8204 8208 8212 8216 82
20 8224 8228 8232 8236 8240"
90 RFA$=RFA$&" 8244 3456 345
2 3720 4048 3460 3444 3724 4
088 4526 4792 3386  11233653
3366033628"
650 CALL PEEK(8194,I,J)
680 IF A<9460 THEN 950
1870 H$=SEG$(C$,P,3)&"0"
1880 GOSUB 3890
1890 C=D
1930 IF A>16383 THEN 950
2190 CALL LOAD(8194,T,A-256*
T)
2820 CALL PEEK(8196,R,T)
2910 CALL LOAD(8196,Z1,A-6-2
56*Z1)
3470 RFP=POS(RF$,Q$,1)
3480 IF RFP=0 THEN 3560
3490 RFP=(RFP-1)/6
3500 RFA=(RFP*5)+1
3510 D=VAL(SEG$(RFA$,RFA,5))
3520 GOTO 3810
3530 REM
3540 REM
3550 REM
3840 H$="" :: Y=4096
3845 FOR I=0 TO 3
3850 H=INT(D/Y)
3855 H$=H$&CHR$(H+48-(7*(H>9
)))
3860 D=D-Y*H
3865 Y=Y/16
3870 NEXT I
3890 D=0 :: Y=1
3895 FOR I=LEN(H$)TO 1 STEP
-1
3900 H=POS("0123456789ABCDEF
",SEG$(H$,I,1),1)
3910 IF H=0 THEN 3560
3920 D=D+(H-1)*Y
3930 Y=Y*16
3940 NEXT I
```

HARD-COPY

Hardcopy ist eigentlich ein alter Hut. Jedoch werden diejenigen unter Ihnen, die zum Beispiel einen Seikosha GP 100 A besitzen, festgestellt haben, daß die Sache mit der Hardcopy bei diesem Druckertyp etwas komplizierter ist als sonst.
Einige von Ihnen werden sicherlich schon einmal versucht haben, eine in Basic erstellte Grafik auf den Drucker zu bringen, um dann festzustellen, daß zu ihrem „Glück" die acht Pixelreihe jedes Zeichens fehlt. Grundsätzlich besteht bei allen Sieben-Nadel-Druckern dieses Problem.
Nun, dieses Programm soll es Ihnen ermöglichen, Grafiken schwarz auf weiß vor sich liegen zu haben.
Die Sache ist eigentlich ganz einfach. Man braucht ja bloß den achten Pixelpunkt der ersten Druckzeile als ersten Pixelpunkt der zweiten Pixelzeile zu nehmen und den siebten und achten Pixelpunkt der zweiten Druckzeile als ersten und zweiten Pixelpunkt der dritten Druckzeile, usw. Das macht man so lange, bis man die ersten sieben Bildschirmzeilen gedruckt hat und dann geht ab der achten Bildschirmzeile alles wieder von vorne los.
So etwas in Basic zu realisieren ist relativ einfach, aber schließlich und endlich hat nicht jeder die Zeit, bis zu dreieinhalb Stunden auf seine Grafik zu warten. Das vorliegende Assemblerprogramm erlaubt es, die gleiche Grafik in ca. 45 Sekunden zu erhalten.
Das Programm ist voll kommentiert und dürfte beim Abtippen keine Schwierigkeiten bereiten.

Hinweise zum Laden des Programms:

Editor/Assembler-Version:
CALL INIT
CALL LOAD
("DSK1.§HARDCOPY")

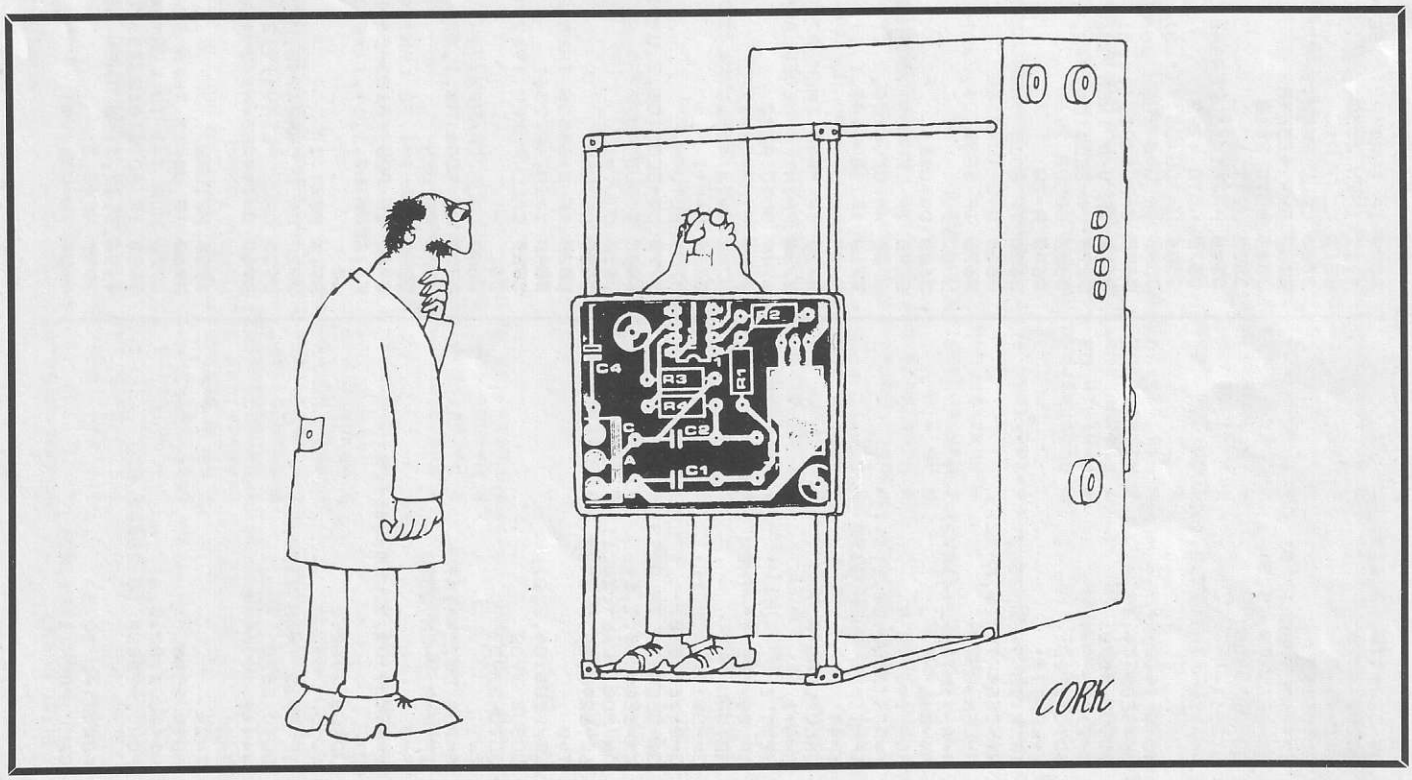
Minimemory-Version:
CALL INIT
CALL LOAD
("DSK1.§HARDCOPY2")

Minimemory-Pokeliste:
OLD CS1
RUN
oder
OLD DSK1.HARDCOPY
RUN

Das Programm wird dann mit CALL LINK ("HCOPY") aufgerufen.

Wenn Sie übrigens das Expanded-Grafik-Basic-Paket von Apesoft besitzen, so können Sie durch Veränderung des folgenden Punktes auch von diesen Grafiken Hardcopys anfertigen.

BASIS DATA >0000
in
BASIS DATA >3000
ändern.
Die Exekution erfolgt ohne jegliche Probleme.
Wie Sie bemerkt haben werden, habe ich keine Hinweise für eine Anpassung des Programms an das Extended-Basic-Modul gegeben. Dies legt daran, daß die DSRLNK- und die GPLLNK-Routine in Extended-Basic nicht vorhanden ist. Wer dennoch auch dort nicht auf eine Hardcopy verzichten möchte, den verweise ich auf dieses Sonderheft. Dort steht eine DSRLNK- und eine GPLLNK-Ersatzroutine für das Extended-Basic-Modul. Bitte versuchen Sie selber, diese Routinen in das Programm einzubauen.
Und nun viel Spaß beim Abtippen. *Bernd Bertling*



```
*      UNL
* HARDCOPY
* ========
*
* (c) by Bernd Bertling, Neu-Crengeldanzstra~e 2
*                        4600 Dortmund 72
*
* Programm f}r Bildschirm-Hardcopy auf einem Seikosha GP 100 A oder anderen
* Druckern mit nur sieben Nadeln.
*
* Notwendige Ger{tekonfiguration:
* -------------------------------
* 1.Editor/Assembler Paket zur Programmerstellung
* 2.Speichererweiterung
* 3.Editor/Assembler- oder Minimemory-Modul
* 4.Diskettensystem mit mind. 1 Laufwerk
* 5.RS232 Schnittstelle
* 6.Seikosha GP 100 A
*
* Programmnamen:  DSK1.@HARDCOPY  --> Name f}r Objekt-Code
* --------------  DSK1.@HARDCOPYS --> Name f}r Source-Code
*
* Aufruf: CALL LINK("HCOPY")
* -------
*
* Programmidentifizierung
*
      IDT  'HARDCOPY'
*
* Die untengenannten Equates m}~en bei Erstellung eines
* Objektcodes f}r das Editor/Assembler-Modul durch folgende
* Refferenzen ersetzt werden.
*
* Refferenzen f}r Editor/Assembler
*
      REF  DSRLNK,GPLLNK,VSBW,VMBW,VSBR,VMBR
*
* Programmdefinition
*
      DEF  HCOPY
*
* Minimemory-Equates
*
DSRLNK EQU  >6038    Verbindung zur DSR-Routine
GPLLNK EQU  >6018    Verbindung zu GROM-Routinen
VSBW   EQU  >6024    Ein Byte aus VDP-RAM lesen
VMBW   EQU  >6028    Mehrere Bytes aus VDP-RAM lesen
VSBR   EQU  >602C    Ein Byte in's VDP-RAM schreiben
VMBR   EQU  >6030    Mehrere Bytes in's VDP-RAM schreiben
*
* sonstige Equates
*
STATUS EQU  >837C    Statusregister
PNTR   EQU  >8356    DSR Namenszeiger
POINT  EQU  >831C    Pab-Pointer
GPLWS  EQU  >83E0    GPL-Arbeitsbereich
*
* Dateidaten f}r den PAB
*
PAB    DATA 800      Auf >0320 im VDP-RAM liegt der Pab, da dort noch
*                    ein paar Bytes frei sind. F}r den Pab-Buffer wird
*                    mittels BLWP @GPLLNK, DATA >0038 Platz reserviert.
PAB5   BSS  2        Buffer f}r PAB-Adresse plus f}nf
PABBUF BSS  2        Buffer f}r PAB-Buffer-Adresse
PDATA  DATA >0012,>0000,>FF00,>0000,>6006
       TEXT 'PIO.CR'
D2     DATA 2
D5     DATA 5
D9     DATA 9
BASIS  DATA >0000       Basisadresse des PDT-Tables in Basic
*
* Eigene Dateidaten
*
MYDAT1 BYTE >0A,>1B,>10,>00,>68  Druckerdaten
MYDAT2 BYTE >0A,>0A,>0A,>0A,>0A  F}nfmal Zeilenvorschub
BYT0   BYTE >00
BYT128 BYTE >80          Bytekonstante
*
* Weitere Dateidaten
*
CLOSE  BYTE 1    Datei schlie~en
WRITE  BYTE 3    Zur Datei schreiben
       EVEN
*
* Der folgende Datenblock enth{lt alle Daten f}r die Auswertung
* der 24 Bildschirmzeilen, welche 28 Druckzeilen entsprechen.
*
* Der Aufbau ist folgenderma~en (Beispiel zweite Druckzeile)
*
* VDP-Adresse f}r erste Spalte und erste Zeile; VDP-Adresse f}r
* erste Spalte und zweite Zeile; Adresse der Variablen, die bearbeitet
* werden soll; Erster Schleifenz{hler f}r 'ZEILE'; Zweiter Schleifen-
* z{hler f}r 'ZEILE'
*
ZLDATA DATA >0000,>0000,STRIN2,>0007
       DATA >0000,>0020,STRIN1+7,>0106
       DATA >0020,>0040,STRIN1+6,>0205
       DATA >0040,>0060,STRIN1+5,>0304
       DATA >0060,>0080,STRIN1+4,>0403
       DATA >0080,>00A0,STRIN1+3,>0502
       DATA >00A0,>00C0,STRIN1+2,>0601
       DATA >00C0,>00C0,STRIN1+1,>0700
       DATA >00E0,>00E0,STRIN2,>0007
       DATA >00E0,>0100,STRIN1+7,>0106
       DATA >0100,>0120,STRIN1+6,>0205
       DATA >0120,>0140,STRIN1+5,>0304
       DATA >0140,>0160,STRIN1+4,>0403
       DATA >0160,>0180,STRIN1+3,>0502
       DATA >0180,>01A0,STRIN1+2,>0601
       DATA >01A0,>01A0,STRIN1+1,>0700
       DATA >01C0,>01C0,STRIN2,>0007
       DATA >01C0,>01E0,STRIN1+7,>0106
       DATA >01E0,>0200,STRIN1+6,>0205
       DATA >0200,>0220,STRIN1+5,>0304
       DATA >0220,>0240,STRIN1+4,>0403
       DATA >0240,>0260,STRIN1+3,>0502
       DATA >0260,>0280,STRIN1+2,>0601
       DATA >0280,>0280,STRIN1+1,>0700
```

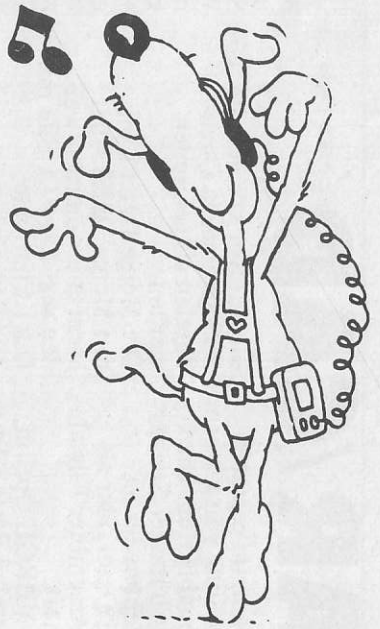
```
       DATA >02A0,>02A0,STRIN2,>0007
       DATA >02A0,>02C0,STRIN1+7,>0106
       DATA >02C0,>02E0,STRIN1+6,>0205
       DATA >02E0,>02E0,STRIN1+5,>0300
*
* Returnlevel f}r verschiedene Unterprogramme
*
BACK1  BSS  2
BACK2  BSS  2
*
* Arbeitsregister
*
MYREGI BSS  32  1.Registersatz
MYREG2 BSS  32  2.Registersatz
REGI   BSS  2   Hilfsregister
REGIHB EQU  REGI
REGILB EQU  REGI+1
*
* Hilfsvariablen
*
HELP1  BSS  2
HELP2  BSS  2
HELP3  BSS  2
MATRIX BSS  56  Bin{rmatrix
STRING BSS  8   Variable f}r 8 Byte langen Seikosha-String
*
* Hilfsvariablen f}r 8 Byte lange TI-Strings
*
STRIN1 BSS  8
STRIN2 BSS  8
*
*
HCOPY  LWPI MYREGI          1.Satz von Arbeitsregistern laden
*
       LI   R0,>00FF        255 Bytes sollen f}r den Pab-Buffer
       MOV  R0,@>830C       reserviert werden.
*
       MOVB @BYT0,@STATUS   GPL-Statusbyte l!schen
       LWPI GPLWS           GPL-Arbeitsregister laden
       BLWP @GPLLNK         Platz im VDP-RAM reservieren
       DATA >0038
*
       LWPI MYREGI          1.Satz von Arbeitsregistern laden
       MOV  @>831A,@PABBUF  Die erste freie Adresse in die Variable
*
       MOV  @PAB,R0         Adresse des Pab nach R0
       MOV  R0,@POINT       Adresse des Pab in den Pab-Pointer schieben
       A    @D5,R0          F}nf hinzuaddieren
       MOV  R0,@PAB5        Adresse in den Buffer schieben
*
       MOV  @PAB,R0         PAB-Adresse nach R0
       LI   R1,PDATA        Dateidaten-Adresse nach R1
       LI   R2,16           16 Bytes sind zu }bertragen
       BLWP @VMBW           Bytes in's VDP-RAM schreiben
*
       MOV  @PAB,R0         An diese Adresse mu~ der Pab-Buffer
       A    @D2,R0
       LI   R1,PABBUF       R1 mit der Adresse der Variablen laden
       LI   R2,2            2 Bytes sollen }bertragen werden
       BLWP @VMBW           Bytes in's VDP-RAM bringen
*
       MOV  @PAB,R6         R6 auf Dateinamen richten
       A    @D9,R6
       BL   @INOUT          Datei er!ffnen
*
       MOVB @WRITE,R1       Ausgabe-Befehl in's PAB bringen
       MOV  @PAB,R0
       BLWP @VSBW
       BL   @INOUT          Datei auf Ausgabe {ndern
*
       LI   R1,>0100        Ein Byte zum Drucker schicken
       MOV  @PAB5,R0
       BLWP @VSBW
*
       MOV  @PABBUF,R0
       LI   R1,>0800        Grafikcode in den PAB-Buffer schreiben
       BLWP @VSBW
       BL   @INOUT          Drucker auf Grafik umstellen
*
       LI   R1,>0500        F}nf Bytes zum Drucker schicken
       MOV  @PAB5,R0
       BLWP @VSBW
*
       MOV  @PABBUF,R0      Den Ausdruck erst ab Position 68 beginnen
       LI   R1,MYDAT1
       LI   R2,5
       BLWP @VMBW
       BL   @INOUT          Daten zum Drucker schicken
*
       LI   R10,28          Druckzeilen-Z{hler
       LI   R8,ZLDATA       Adresse der Zeilenbearbeitungsdaten
*
NEXT1  LI   R15,0           R15=0
       MOV  *R8,R13         1.VDP-Adresse nach R13
       INCT R8              Adresse=Adresse+2
       MOV  *R8,R14         2.VDP-Adresse nach R14
       INCT R8              Adresse=Adresse+2
       MOV  *R8,@HELP1      Derzeitige Variablen-Adresse nach HELP1
       INCT R8              Adresse=Adresse+2
       MOVB *R8,@HELP2      1.Schleifenz{hler nach HELP2
       INC  R8              Adresse=Adresse+1
       MOVB *R8,@HELP3      2.Schleifenz{hler nach HELP3
       INC  R8              Adresse=Adresse+1
*
NEXT2  MOV  R13,R0          1.VDP-Adresse nach R0 kopieren
       INC  R13             VDP-Adresse um eins erh!hen
       BLWP @VSBR           Zeichen aus Namensliste lesen
       SRL  R1,8            Byte in Wort verwandeln
       SLA  R1,3            Zeichen mal acht
       MOV  R1,R0           Definitionslisten-Adresse nach R0 kopieren
       MOV  @BASIS,R1       P.D.T.-Adresse nach R1
       A    R1,R0           Plus Definitionsbasisadresse
       LI   R1,STRIN1       Adresse f}r ersten Acht-Byte-Buffer laden
       LI   R2,8            Acht Bytes sind zu lesen
       BLWP @VMBR           Bytes aus VDP-RAM lesen
       MOV  R14,R0          2.VDP-Adresse nach R0 kopieren
       INC  R14             VDP-Adresse um eins erh!hen
       BLWP @VSBR           Zeichen aus Namensliste lesen
```

```
       SRL   R1,8            Byte in Wort verwandeln
       SLA   R1,3            Zeichen mal acht
       MOV   R1,RØ           Definitionslisten-Adresse nach RØ kopieren
       MOV   @BASIS,R1       P.D.T-Adresse nach R1
       A     R1,RØ           Plus Definitionsbasisadresse
       LI    R1,STRIN2       Adresse f}r zweiten Acht-Byte-Buffer laden
       LI    R2,8            Acht Bytes sind zu lesen
       BLWP  @VMBR           Bytes aus VDP-RAM lesen
       LWPI  MYREG2          2.Satz von Arbeitsregistern laden
       CLR   R1              R1=Ø
       LI    R15,MATRIX      R15 mit der Matrixadresse laden
       MOV   @HELP1,R14      Aktuelle STRIN1- oder STRIN2-Adresse gewinnen
       LI    R13,STRIN2      R13 mit STRIN2-Adresse laden
       MOVB  @HELP2,R9       Ersten Schleifenz{hlerwert laden
       MOVB  R9,R9           Soll die Schleife ausgef}hrt werden ?
       JEQ   NEXT4           Nein, also n{chste Schleife
*
       SWPB  R9              Ja, also Wort in Byte verwandeln
       MOVB  R1,R9           MS-Byte von R9 l!schen
*
NEXT3  LI    RØ,7            RØ mit Verschiebewert laden
       MOVB  *R14,R5         Byte nach R5 kopieren
       MOVB  R5,R7           Byte in R7 sichern
       BL    @COPIER         Byte in Bitmatrix eintragen
       INC   R14             R14=R14+1
       DEC   R9              Schleifenz{hler um eins erniedrigen
       JNE   NEXT3           Wenn Schleife noch nicht fertig, n{chstes Byte
*
NEXT4  MOVB  @HELP3,R9       Zweiten Schleifenz{hlerwert laden
       MOVB  R9,R9           Soll die Schleife ausgef}hrt werden ?
       JEQ   NEXT6           Nein, also Bitmatrix auswerten
*
       SWPB  R9              Ja, also Wort in Byte verwandeln
       MOVB  R1,R9           MS-Byte von R9 l!schen
*
NEXT5  LI    RØ,7            RØ mit Verschiebewert laden
       MOVB  *R13,R5         Byte nach R5 kopieren
       MOVB  R5,R7           Byte in R7 sichern
       BL    @COPIER         Byte in Bitmatrix eintragen
       INC   R13             R13=R13+1
       DEC   R9              Schleifenz{hler um eins erniedrigen
       JNE   NEXT5           Wenn Schleife noch nicht fertig, n{chstes Byte
*
NEXT6  BL    @DSGNCH         Bitmatrix auswerten
       BL    @BRING          Seikosha-String in den Pab-Buffer bringen
       INC   R15             Spaltenz{hler um eins erh!hen
       CI    R15,32          Eine Zeile ausgewertet ?
       JNE   NEXT2           Nein, also n{chste Spalte
*
       BL    @DRUCK          Eine Zeile fertig, also ausdrucken
       DEC   R1Ø             Druckzeilenz{hler um eins erniedrigen
       JNE   NEXT1           Noch nicht alle Zeilen gedruckt
*
       LI    R1,>Ø5ØØ        Alle Zeilen gedruckt, f}nf Bytes zum Drucker
       MOV   @PAB5,RØ
       BLWP  @VSBW
*
       MOV   @PABBUF,RØ      Daten f}r Zeilenvorschub laden
       LI    R1,MYDAT2
       LI    R2,5
       BLWP  @VMBW
       BL    @INOUT          Zeilenvorschub ausf}hren
*
       LI    R1,>Ø1ØØ        Eintrittspunkt f}r vorzeitigen Programmabbruch
       MOV   @PAB5,RØ        Ein Byte zum Drucker schicken
       BLWP  @VSBW
*
       MOV   @PABBUF,RØ
       LI    R1,>ØF ØØ       Textmoduscode in den PAB-Buffer schreiben
       BLWP  @VSBW
       BL    @INOUT          Drucker wieder auf Standardmodus umschalten
*
       LI    R1,>ØØØØ        Aus PAB-Buffer kein Byte mehr zum Drucker schicken
       MOV   @PAB5,RØ
       BLWP  @VSBW
*
       MOVB  @CLOSE,R1       Datei-Schlie~-Befehl in's PAB bringen
       MOV   @PAB,RØ
       BLWP  @VSBW
       BL    @INOUT          Datei sclie~en
*
       MOVB  @BYTØ,@STATUS   Statusregister l!schen
       LWPI  GPLWS           GPL-Arbeitsregister laden
       B     @>ØØ7Ø          N{chste Basic-Anweisung ausf}hren
*
* Unterprogramme
* **************
*
* BRING
*
* Diese Unterprogramm bringt den neuen Seikosha-String
* in den PAB-Buffer.
*
BRING  MOV   R11,@BACK1      Returnadresse sichern
       MOV   @PABBUF,RØ      RØ mit der Adresse des Pabbuffer's laden
       MOV   R15,R12         Wert von R15 nach R12 kopieren
       SLA   R12,3           R12=Wert von R12 mal 8
       A     R12,RØ          R12=R12+Adresse von PABBUF
       LI    R1,STRING       R1 mit Adresse von STRING laden
       LI    R2,8            Acht Byte sind zu }bertragen
       BLWP  @VMBW           Byte's in das VDP-RAM schreiben
*
       MOV   @BACK1,R11      Returnadresse wiedergewinnen
       RT                    Zur}ck zum rufenden Programm
*
* DRUCK
*
* Wenn eine komplette Bildschirmzeile f}r den Drucker
* aufbereitet wurde, wird sie durch dieses Unterprogramm
* ausgedruckt und der Drucker f}r die n{chste Zeile vor-
* bereitet.
*
DRUCK  MOV   R11,@BACK1      Returnadresse sichern
       LI    R1,>FFØØ        256 Bytes sollen zum Drucker geschickt werden
       MOV   @PAB5,RØ
       BLWP  @VSBW
       BL    @INOUT          Daten zum Drucker schicken
*
```

```
       LI   R1,>0500    F}nf Bytes sollen zum Drucker geschickt werden
       MOV  @PAB5,RØ
       BLWP @VSBW
*
       MOV  @PABBUF,RØ  Daten f}r Druckposition in den PAB-Buffer laden
       LI   R1,MYDAT1
       LI   R2,5        F}nf Bytes sind zu }bertragen
       BLWP @VMBW       Bytes in's VDP-RAM schreiben
       BL   @INOUT      Daten zum Drucker schicken
*
       MOV  @BACK1,R11  Returnadresse wiedergewinnen
       RT               Zur}ck zum rufenden Programm
*
* DSGNCH
*
* Mit Hilfe dieses Unterprogramms wird aus der Bitmatrix der neue
* Seikosha-String berechnet.
*
DSGNCH MOV  R11,@BACK1  Returnadresse sichern
       CLR  RØ          RØ=Ø
       LI   R1,STRING   R1 mit Adresse von STRING laden
       LI   R2,4        4 mal soll die folgende Schleife durchlaufen werden
NEXT7  MOV  RØ,R1       Ein Wort l{schen
       INCT R1          Adresse in R1 um zwei erh{hen
       DEC  R2          Schleifenz{hler um eins erniedrigen
       JNE  NEXT7       Wenn noch nicht alles gel{scht ist n{chstes Wort
*
       LI   R9,8        Schleifenz{hler f}r ein Byte
       LI   R12,STRING  R12=Adresse von STRING
       LI   R15,MATRIX  R15=Adresse der Bitmatrix
       LI   R4,8        Z{hlkonstante f}r aktuelle Bitmatrixadresse
*
NEXT8  MOV  R15,R1Ø     Bitmatrixadresse nach R1Ø kopieren
       LI   R11,6       Z{hler f}r zu berechnende Bits
       LI   RØ,1        RØ mit Verschiebewert laden
       CLR  R8          R8=Ø
       MOVB *R1Ø,R8     Bit Ø von der Bitmatrix nach R8 kopieren
       A    R4,R1Ø      Aktuelle Bitmatrixadresse um acht erh{hen
*
NEXT9  MOVB *R1Ø,R13    N{chstes Bit nach R13 bringen
       SWPB R13         Byte in Wort verwandeln
       SLA  R13,Ø       R13 wird nun um Wert von RØ nach links verschoben
*                       --> R13 mal 2^Wert von RØ
       SWPB R13         Wort in Byte verwandeln
       AB   R13,R8      Die einzelnen Bits zu einem Byte aufaddieren
       A    R4,R1Ø      Aktuelle Bitmatrixadresse um acht erh{hen
       INC  RØ          Verschiebewert um eins erh{hen
       DEC  R11         Bitz{hler um eins erniedrigen
       JNE  NEXT9       Wenn noch keine sieben Bits berechnet, n{chstes Bit
*
       AB   @BYT128,R8  Da das achte Bit bei dem Seikosha GP 1ØØ A immer
*                       auf 1 sein mu~, werden hier 128 zu dem Byte hinzu-
*                       addiert.
       MOVB R8,*R12     Fertiges Byte in den String-Buffer packen
       INC  R15         Urspr}ngliche Bitmatrixadresse um eins erh{hen
       INC  R12         String-Bufferadresse um eins erh{hen
       DEC  R9          Sind acht Bytes berechnet ?
       JNE  NEXT8       Nein, also n{chstes Byte berechnen
*
       LWPI MYREGI      Ja, also wieder die Arbeitsregister f}r das
*                       Hauptprogramm laden
       MOV  @BACK1,R11  Returnadresse wiedergewinnen
       RT               Zur}ck zum rufenden Programm
*
* COPIER
*
* Erkl{rungen siehe Unterprogramm 'ZEILE'.
*
COPIER MOV  R11,@BACK2  Returnadresse sichern
*
NEXT1Ø SRL  R5,8        Byte in Wort verwandeln
       SRL  R5,Ø        R5 um Wert von RØ verschieben(dividieren)
       SWPB R5          Wort in Byte verwandeln
       MOVB R5,*R15+    Bit in die Bitmatrix schreiben und Adresse + eins
       SWPB R5          Byte in Wort verwandeln
       SLA  R5,Ø        R5 um Wert von RØ verschieben(multiplizieren)
       SWPB R5          Wort in Byte verwandeln
       SB   R5,R7       Aktuellen Restwert gewinnen
       MOVB R7,R5       Restwert nach R5 kopieren
       DEC  RØ          Verschiebewert um eins erniedrigen
       JNE  NEXT1Ø      Wenn Byte noch nicht zerlegt n{chstes Bit
*
       MOVB R5,*R15+    Bit 7 kann nur noch 1 oder Ø sein
*
       MOV  @BACK2,R11  Returnadresse wiedergewinnen
       RT               Zur}ck zum rufenden Programm
*
* INOUT
*
* File-Operation durchf}hren
*
INOUT  MOV  R6,@PNTR    L{ngenzeiger setzen
       MOVB @BYTØ,@STATUS GPL-Statusbyte l{schen
*
       MOVB @>98Ø2,@REGIHB Zuerst High-Byte und
       MOVB @>98Ø2,@REGILB dann Low-Byte sichern
       DEC  @REGIHB     High-Byte minus eins
*
       BLWP @DSRLNK     File-Operation durchf}hren
       DATA 8
*
       MOVB @REGIHB,@>9CØ2 Zuerst High-Byte und
       MOVB @REGILB,@>9CØ2 dann Low-Byte positionieren
*
       RT               Zur}ck zum rufenden Programmteil
*
* Ende des Programms
*
       END
```

```
100 REM Hardcopyprogramm
          f}r
        SEIKOSHA GP 100 A
110 REM
120 REM POKELISTE f}r das
        Minimemory-Modul
130 REM (c) by
    Bernd Bertling
    Neu-Crengeldanzstra~e 2
    4600 Dortmund 72
140 REM
150 REM Belegter Speicher-
        platz:
        >7118 --> >75AD
        28952 --> 30125
160 DATA 3,32,0,0,0,0,0,18,0
,0,255,0,0,0,96,6
170 DATA 80,73,79,46,67,82,0
,2,0,5,0,9,0,0,10,27
180 DATA 16,0,104,10,10,10,1
0,10,0,128,1,3,0,0,0,0
190 DATA 114,184,0,7,0,0,0,3
2,114,183,1,6,0,32,0,64
200 DATA 114,182,2,5,0,64,0,
96,114,181,3,4,0,96,0,128
210 DATA 114,180,4,3,0,128,0
,160,114,179,5,2,0,160,0,192
220 DATA 114,178,6,1,0,192,0
,192,114,177,7,0,0,224,0,224
230 DATA 114,184,0,7,0,224,1
,0,114,183,1,6,1,0,1,32
240 DATA 114,182,2,5,1,32,1,
64,114,181,3,4,1,64,1,96
250 DATA 114,180,4,3,1,96,1,
128,114,179,5,2,1,128,1,160
260 DATA 114,178,6,1,1,160,1
,160,114,177,7,0,1,192,1,192
270 DATA 114,184,0,7,1,192,1
,224,114,183,1,6,1,224,2,0
280 DATA 114,182,2,5,2,0,2,3
2,114,181,3,4,2,32,2,64
290 DATA 114,180,4,3,2,64,2,
96,114,179,5,2,2,96,2,128
300 DATA 114,178,6,1,2,128,2
,128,114,177,7,0,2,160,2,160
310 DATA 114,184,0,7,2,160,2
,192,114,183,1,6,2,192,2,224
320 DATA 114,182,2,5,2,224,2
,224,114,181,3,0,0,0,0,0
330 DATA 2,224,114,40,2,0,0,
255,200,0,131,12,216,32,113,
64
340 DATA 131,124,2,224,131,2
24,4,32,96,24,0,56,2,224,114
,40
350 DATA 200,32,131,26,113,2
8,192,32,113,24,200,0,131,28
,160,32
360 DATA 113,48,200,0,113,26
,192,32,113,24,2,1,113,30,2,
2
370 DATA 0,16,4,32,96,40,192
,32,113,24,160,32,113,46,2,1
380 DATA 113,28,2,2,0,2,4,32
,96,40,193,160,113,24,161,16
0
390 DATA 113,50,6,160,117,12
8,208,96,113,67,192,32,113,2
4,4,32
400 DATA 96,36,6,160,117,128
,2,1,1,0,192,32,113,26,4,32
410 DATA 96,36,192,32,113,28
,2,1,8,0,4,32,96,36,6,160
420 DATA 117,128,2,1,5,0,192
,32,113,26,4,32,96,36,192,32
430 DATA 113,28,2,1,113,54,2
,2,0,5,4,32,96,40,6,160
440 DATA 117,128,2,10,0,28,2
,8,113,68,2,15,0,0,195,88
450 DATA 5,200,195,152,5,200
,200,24,114,106,5,200,216,24
,114,108
460 DATA 5,136,216,24,114,11
0,5,136,192,13,5,141,4,32,96
,44
470 DATA 9,129,10,49,192,1,1
92,96,113,52,160,1,2,1,114,1
76
480 DATA 2,2,0,8,4,32,96,48,
192,14,5,142,4,32,96,44
490 DATA 9,129,10,49,192,1,1
92,96,113,52,160,1,2,1,114,1
84
500 DATA 2,2,0,8,4,32,96,48,
2,224,114,72,4,193,2,15
510 DATA 114,112,195,160,114
,106,2,13,114,184,210,96,114
,108,210,73
520 DATA 19,11,6,201,210,65,
2,0,0,7,209,94,209,197,6,160
530 DATA 117,94,5,142,6,9,22
,247,210,96,114,110,210,73,1
9,11
540 DATA 6,201,210,65,2,0,0,
7,209,93,209,197,6,160,117,9
4
550 DATA 5,141,6,9,22,247,6,
160,116,254,6,160,116,164,5,
143
560 DATA 2,143,0,32,22,177,6
,160,116,196,6,10,22,158,2,1
570 DATA 5,0,192,32,113,26,4
,32,96,36,192,32,113,28,2,1
580 DATA 113,59,2,2,0,5,4,32
,96,40,6,160,117,128,2,1
590 DATA 1,0,192,32,113,26,4
,32,96,36,192,32,113,28,2,1
600 DATA 15,0,4,32,96,36,6,1
60,117,128,2,1,0,0,192,32
610 DATA 113,26,4,32,96,36,2
08,96,113,66,192,32,113,24,4
,32
620 DATA 96,36,6,160,117,128
,216,32,113,64,131,124,2,224
,131,224
630 DATA 4,96,0,112,200,11,1
14,36,192,32,113,28,195,15,1
0,60
640 DATA 160,12,2,1,114,168,
2,2,0,8,4,32,96,40,194,224
650 DATA 114,36,4,91,200,11,
114,36,2,1,255,0,192,32,113,
26
660 DATA 4,32,96,36,6,160,11
7,128,2,1,5,0,192,32,113,26
670 DATA 4,32,96,36,192,32,1
13,28,2,1,113,54,2,2,0,5
680 DATA 4,32,96,40,6,160,11
7,128,194,224,114,36,4,91,20
0,11
690 DATA 114,36,4,192,2,1,11
4,168,2,2,0,4,192,64,5,193
700 DATA 6,2,22,252,2,9,0,8,
2,12,114,168,2,15,114,112
710 DATA 2,4,0,8,194,143,2,1
1,0,6,2,0,0,1,4,200
720 DATA 210,26,162,132,211,
90,6,205,10,13,6,205,178,13,
162,132
730 DATA 5,128,6,11,22,247,1
78,32,113,65,215,8,5,143,5,1
40
740 DATA 6,9,22,232,2,224,11
4,40,194,224,114,36,4,91,200
,11
750 DATA 114,38,9,133,9,5,6,
197,223,197,6,197,10,5,6,197
760 DATA 113,197,209,71,6,0,
22,245,223,197,194,224,114,3
8,4,91
770 DATA 200,6,131,86,216,32
,113,64,131,124,216,32,152,2
,114,104
780 DATA 216,32,152,2,114,10
5,6,32,114,104,4,32,96,56,0,
8
790 DATA 216,32,114,104,156,
2,216,32,114,105,156,2,4,91,
0,0
800 CALL INIT
810 REM
820 REM Schreibschleife
830 REM
840 REM Die Variable 'T'
soll Ihnen eine Sicherheit
geben, die Data's richtig
eingegeben zu haben.
850 REM Als Pr}fsumme mu~
        73194 herauskommen.

860 FOR X=28952 TO 29223 STE
P 16
870 READ A,B,C,D,E,F,G,H,I,K
,L,M,N,O,P,Q
880 T=T+A+B+C+D+E+F+G+H+I+K+
L+M+N+O+P+Q
890 CALL LOAD(X,A,B,C,D,E,F,
G,H,I,K,L,M,N,O,P,Q)
900 NEXT X
910 FOR X=29376 TO 30127 STE
P 16
920 READ A,B,C,D,E,F,G,H,I,K
,L,M,N,O,P,Q
930 T=T+A+B+C+D+E+F+G+H+I+K+
L+M+N+O+P+Q
940 CALL LOAD(X,A,B,C,D,E,F,
G,H,I,K,L,M,N,O,P,Q)
950 NEXT X
960 REM Def-Pointer setzen
970 CALL LOAD(28700,113,24,1
27,248)
980 REM Def-Table setzen
990 CALL LOAD(32760,72,67,79
,80,89,32,114,192)
1000 END
```



CAR/CAD

Das Programm gestattet die Erstellung von kurzen Trickfilmabläufen. Die erstellten Character Strings werden entweder als MERGE Datei oder als Editor/Assembler File abgespeichert.
Mit der MERGE Option des Extended Basic können die erstellten Strings dann in beliebige Basic-Programme übernommen werden. Mit dem Assembler/Editor können die Strings auch in Quelldateien eingebunden werden.
Die Merge Datei kann mit Merge DSK1.Char Mem übernommen werden und das E/A File mit Dsk1.Charcode.
Zur Bedienung:
Nachdem das Basicprogramm geladen und gestartet ist, beginnt der Bildschirmaufbau. Zuerst läuft eine kleine Demonstration ab. Danach wird das Zeichenfeld gelöscht und es erscheint der Sprite/Cursor.
Wird jetzt der Joystick nicht benutzt, und auch keine Taste gedrückt, springt das Zeichenprogramm "Draw" selbsttätig in das Basicprogramm zurück.
Die Geschwindigkeit, mit der sich der Cursor bewegt und die Zeit, die vergeht bis das Programm zurückspringt, wird im Call Link Aufruf festgelegt.
Das "Select" Programm ermöglicht jetzt die Auswahl eines Befehles.
Das "Select" Programm springt erst bei Betätigung der Feuertaste wieder zum Basic Programm zurück.
Sowohl bei "Draw" als auch bei "Select" werden bei einem Rücksprung zum Basic-Programm die aktuelle Zeile und Spalte der Cursorposition an das aufrufende Programm übergeben.
Die Programmsegmente "Draw", "Select" und "Scroll" sind außerdem ohne weitere Probleme auch in andere Programme einzubinden.
Ich hoffe, daß dies als Anleitung zum Programm genügt, wenn man ein wenig mit dem Programm spielt, versteht man es wohl am schnellsten.

Eberhard Schael

Erklärungen zu den Programmsegmenten:

@CCAD/DEF#
Enthält alle für das Maschinenprogramm wichtigen Variablenspeicher, REF/DEF Table, Zeichendefinitionen sowie die Copy Direktiven für die restlichen Files.

@CHARSCAN#
Dieses Programmsegment tastet das Zeichenfeld ab und liefert die Variable VAR$ an das TI-Basic-Programm zurück.
Der Aufruf aus dem Basic erfolgt mit Call Link (CHRSCN",VAR$).

@DSPACPT#
Entspricht dem Programm aus 99 Special II, jedoch wurde z.B. auf modulunabhängige Routinen verzichtet.
Aufruf über
Call Link ("DSPLAY", Zeile, Spalte, Stringlänge, VAR§)
Call Link ("ACCEPT", Zeile, Spalte, Stringlänge, VAR$)

@INVERT#
Tauscht im Zeichenfeld das Leerzeichen (ASCII 32) gegen dieses Zeichen (ASCII 126), das ist die Tilde (~), aus, und umgekehrt.
Aufruf über Call Link ("INVERT")

@SCROLL#
Diese Routine scrollt ein im Call Link Statement definiertes Feld in eine von 4 vorgegebenen Richtungen.
Aufruf (auch aus anderen TI-Basic-Programmen) über Call Link ("SCROLL", ZEMIN; SPMIN, ZEMAX, SPMAX, Richtung, Scrollcount)
Das durch ZEMIN, SPMIN, ZEMAX, SPMAX definierte Feld wird nach unten =1, nach oben =2, nach links =3 und nach rechts =4 gescrollt. Scrollcount gibt an, wie oft dieser Vorgang wiederholt wird.

@SCRMEMO#
Call Link ("SCRSAV")
Diese Routine speichert beim Aufruf den kompletten Bildschirminhalt.
Call Link ("SCRRET")
Bringt den zuvor gespeicherten Bildschirm wieder zurück.

@SELECT#
Diese Routine bewegt in dem – im Call Link Statement – definierten Zeichenfeld den Cursor. Sobald die Feuertaste gedrückt wird, springt die Routine zurück zum Basic-Programm.
Die augenblickliche Position des Cursors (Zeile + Spalte) wird dabei den Variablen Value 1 und Value 2 zugewiesen.
Aufruf über
Call Link ("SELECT", ZEMIN, SPMIN,

TI 99/4A

PHERIPHERIE

Discontroller (Orig. TI)	399,–
RS 232 Karte (Orig. TI)	399,–
RS 232 Karte (Atronic)	359,–
P-Code-Karte (Orig. TI)	749,–
32 K-Karte (Atronic)	379,–
Discontroller DSDD (Atronic)	489,–
Discontroller DSDD (Corcomp)	629,–
Compact Peripherie System CPS 99 mit 1 Diskettenlaufwerk DSDD + 10 Disketten	1598,–
Diskettenlaufwerk intern DSDD (Epson) mit Einbausatz	429,–
Externe 256 K-Erweiterung	589,–
Externe 32 K-Erweiterung	239,–
dto. + 1 Centronicsschnittst.	289,–
Externe 32 K-Erweiterung + Centronicsschnittstelle + Kabel + Epsondrucker LX 80	1259,–
dto. + Epsondrucker FX 85	1759,–
dto. + Stardrucker SG 10	1279,–
Sprachsynthesizer	189,–
Modulexpander 3fach	125,–
RGB-Modulator	179,–
Akustikkoppler Dataphon S 21 d + externe V-24-Schnittstelle + Verbindungskabel	559,–
TI-Maus anschlußfertig	295,–
Joystickinterface + 2 Joysticks Quickshot II	89,–
Cassettenrecorderkabel	29,–
MBX-Sprachsteuereinheit + Baseballmodul anschlußfertig	349,–
Grafiktablett Supersketch + Dig Dug + Defender + Statistik	199,–

BÜCHER

Editor/Assembler Handbuch dt.	98,–
TI-Basic & Extended Basic dt.	48,–
Mini Memory Spezial dt.	55,–
TI-99/4 A Intern dt.	38,–
TMS 9900 Assemblerhandbuch für das Mini Memory dt.	78,–
TI-99/4 A Intern dt.	38,–

MODULSOFTWARE

Extended Basic (dt. Nachbau)	199,–
Extended Basic II Plus	289,–
Mini Memory + Assemblerhandbuch Mini Memory dt.	269,–
Editor/Assembler (32 K notw.)	179,–
TI-Writer (32 K notw.)	299,–
TI-Logo II (32 K notw.)	299,–
Miltiplan (32 K notw.)	259,–
Diskfixer (Navarone)	149,–
Terminal Emulator II	85,–
Connect four, Yahtzee, Attack Alpiner, Car Wars, Chisholm Trail, Othello, Invaders, Munch Man	je 39,–
Blackjack, Fathom, Hopper, Dig Dug, Defender, Soccer, Parsec	je 49,–
Burgertime, Congo Bongo, Espial, Moonsweeper, Treasure Island, Microsurgeon, Bigfoot, Statistik	je 59,–
Star Trek, Tunnels of doom, Touch Typing Tutor	je 69,–
Buck Rogers, Return to Pirat's Isle, Adventuremodul, Video Chess	je 75,–
Datenverwaltung + Analyse	79,–
Popeye, Jungle Hunt, Moon Patrol, Ms. Pacman, Pole Position, Donkey Kong, Protector II, Shamus	je 89,–
Video Chess + Defender + Dig Dug	nur 129,–
Alpiner + Munch Man + Microsurgeon	nur 119,–

DISKETTEN- UND CASSETTENSOFTWARE

Superbasic, Exbasic II + Painter, Extended Basic Compiler, Graphicmaster, 3D-World, Forth, Apesoft-Programme, Flugsimulation, Skat, „Der schwarze Kristall" a.A.

Alle Preise inkl. MwSt. zus. Versandkostenpauschale (Warenwert bis DM 1000,–/darüber): Vorauskasse (DM 8,–/20,–), Nachnahme (DM 11,20/23,20). Ausland (DM 18,–/30,–). Versand nur gegen Vorauskasse oder per NN; Ausland nur gegen Vorauskasse. Gesamtpreisliste gegen Freiumschlag.

CSV RIEGERT
Schloßhofstr. 5, 7324 Rechborghausen, Tel. (07161) 5 28 89

ZEMAX, SPMAX, ZEPOS, SSPOS, Delaytime, Value 1, Value 2).

@**DRAW**#
Aufruf der Routine über Call Link ("DRAW", ZEMIN, SPMIN, ZEMAX, SPMAX, ZEPOS, SPPOS, Delaytime, Returntime, Value 1, Value 2).
Wenn die Routine zurückspringt in das Basic-Programm wird die Position des Cursors (Zeile + Spalte) an die Variablen Value 1 + Value 2 übermittelt.

@**DRAW**#
Dieses ist die Zeichenroutine.
Aufruf über Call Link ("DRAW", ZEMIN, SPMIN, ZEMAX, SPMAX, ZEPOS, SPPOS, Delaytime, Returntime)

Die ersten vier Parameter definieren auch hier wieder das Zeichenfeld. ZEPOS & SPPOS geben die Position an, wo bei Aufruf der Routine ein Sprite erscheint.
Dieses Sprite ist ein Cursor, der die augenblickliche Schreibposition markiert.

Bewegt wird Cursor mit dem Joystick 1. Wird die Feuertaste gedrückt, dann erscheint an dieser Stelle eine Tilde (ASCII 126). Ohne die Feuertaste zu drücken, löscht das Kreuz die betreffende Schreibposition.
Delaytime bestimmt die Geschwindigkeit des Cursors und kann zwischen 1 – 2000 liegen.
Werden Joystick und Feuertaste nicht benutzt, springt die Zeichenroutine bei Erreichen von Returntime zurück in das Basic-Programm.

@**SUBPROG**#
Dieses Segment enthält alle mehrfach benutzten Routinen.

Zum Basic-Programm CHAR/CAD:

Das Programm ermöglicht das Erstellen von Charactercodes für max. 4 Schreibstellen gleichzeitig. Springt der Cursor aus dem Zeichenfeld zur Auswahlliste, wählt man mit dem Joystick den entsprechenden Befehl. Sobald die Zeichenroutine wieder ins Basic springt (keine Taste drücken, Joystick 1 nicht bewegen!), wird der entsprechende Befehl ausgeführt.
Da es lästig ist, gespeicherte Characterstrings abzutippen oder als Datei in Programme einzubinden, speichert das Programm die erstellten Strings als Merge-Datei unter dem Namen "CHARMEM" auf der Diskette.
Unter Zuhilfenahme des EX-Basic-Moduls kann die Datei – durch Merge "DSK 1.CHARMEM" – in ein anderes Programm übernommen werden.
Außerdem ermöglicht der Befehl "TRICKFILM" kleine Bildfolgen mit Trickfilmcharacter, sofern mehrere Codes erstellt werden.

Anmerkung: Zu dem Programm müssen die Utilities BSCSUB des E/A Moduls geladen werden.

Benötigte Ausstattung:

Konsole, E/A Modul, P-Box mit 32 K RAM, 1 Diskettenstation, Joystick 1 und EX-BASIC Modul

```
**************************************
*    EBERHARD SCHAEL                 *
*    HAGENER STR. 15                 *
*    5800 HAGEN 1                    *
*    02331/687154                    *
**************************************
*SPROGRAMMSEGMENT "@CCAD/DEF#"
*LAENGE=129 ZEILEN
*
        DEF  DRAW,CHRSCN,INVERT
        DEF  DSPLAY,ACCEPT,SCROLL
        DEF  SCRSAV,SCRRET,SELECT
*
        REF  VSBW,VMBW,VMBR,VSBR
        REF  KSCAN,STRASG,STRREF
        REF  NUMASG,NUMREF,ERR,XMLLNK
*
MYWS    BSS  >20      EIGENE ARBEITSREGISTER
RØHB    EQU  MYWS     ABSOLUTE ADRESSEN FUER
RØLB    EQU  MYWS+1   ARBEITSREGISTER RØ
*
        EVEN
STRING  BSS  >41      BUFFER FUER CHARACTERSTRING
DASTRG  BSS  >21      BUFFER FUER DISPLAY&ACCEPT STRING
SCREEN  BSS  >300     BUFFER FUER EINEN SCHIRMINHALT =768 CHARACTERS
*
CHRBUF  BSS  4        BUFFER UND FLAGS FUER
FLAGØ1  BSS  2        DAS PROGRAMMSEGMENT
FLAGØ2  BSS  2        CHARSCAN
CHRBYT  BSS  2
WRDBUF  BSS  2
SCRCHR  BSS  1
SCRBUF  BSS  32
*
ROW     BSS  >2       BUFFER FUER ZEILE
COL     BSS  >2       BUFFER FUER SPALTE
PRMTR   BSS  >2       LINK PARAMETER BUFFER
ZEMAX   BSS  >2       BUFFER FUER DIE HOECHSTE ZEILE
ZEMIN   BSS  >2       "    "   "   NIEDRIGSTE ZEILE
SPMAX   BSS  >2       "    "   "   HOECHSTE SPALTE
SPMIN   BSS  >2       "    "   "   NIEDRIGSTE SPALTE
*
RTLEV1  BSS  >2       DIVERSE BUFFER FUER
RTLEV2  BSS  >2       RETURNADRESSEN IN VER-
RTLEV3  BSS  >2       SCHIEDENEN LEVELN
TACT    BSS  >2       BUFFER FUER Time•ACTu•ll
NEWPOS  BSS  >2       BUFFER FUER NEUE POSITION
ZEPOS   BSS  >2       BUFFER FUER ZEILE UND
SPPOS   BSS  >2       SPALTE
DLYTME  BSS  >2       BUFFER FUER DELAYTIME
TVRG    BSS  >2       BUFFER FUER VORGABEZEIT
SCRDIR  BSS  >2       BUFFER FUER SCROLL-RICHTUNG
SPDIF   BSS  >2       BUFFER FUER DIFFERENZBETRAG AUS SPMAX UND SPMIN
ZEDIF   BSS  >2       "    "   "   "   "   "   ZEMAX  "   ZEMIN
SCRCNT  BSS  >2       BUFFER FUER SCROLLCOUNT
*
H00     DATA 0000     WORTKONSTANTE
H29     DATA 29
H30     DATA 30
```

```
CHAR00 DATA >0000
VALUE1 DATA >0001
VALUE2 DATA >0002
*
*HEXDTA ENTHAELT DIE MOEGLICHEN KOMBINATIONEN
*FUER DIE ABTASTUNG DES ZEICHENFELDES.
*
HEXDTA DATA >0000,>0000,>0000,>00DE
       DATA >0000,>DE00,>0000,>DEDE
       DATA >00DE,>0000,>00DE,>00DE
       DATA >00DE,>DE00,>00DE,>DEDE
       DATA >DE00,>0000,>DE00,>00DE
       DATA >DE00,>DE00,>DE00,>DEDE
       DATA >DEDE,>0000,>DEDE,>00DE
       DATA >DEDE,>DE00,>DEDE,>DEDE
*
*HXBYTE ENTHAELT DIE ASCII-WERTE FUER 0-9&A-F
*
HXBYTE BYTE >30,>31,>32,>33,>34,>35
       BYTE >36,>37,>38,>39,>41,>42
       BYTE >43,>44,>45,>46
*
BYT1   BYTE 1       BYTE KONSTANTE
BYT28  BYTE 28
BYT96  BYTE 96
BYTM96 BYTE -96
*
       EVEN
JYUP   BYTE  4,0    *JOYST HOCH
JYRT   BYTE  0,4    *JOYST RECHTS
JYDN   BYTE  -4,0   *JOYST RUNTER
JYLT   BYTE  0,-4   *JOYST LINKS
*
JYLTDN BYTE  -4,-4 *JOYST LINKS&RUNTER
JYLTUP BYTE  4,-4  *JOYST LINKS&HOCH
JYRTDN BYTE  -4,4  *JOYST RECHTS&RUNTER
JYRTUP BYTE  4,4   *JOYST RECHTS&HOCH
*
HEXFF  BYTE  >FF     KEINE TASTE
KEY18  BYTE  18      TASTE Q ODER FEUERKNOPF
CLEAR  BYTE  2         "   CLEAR
ERASE  BYTE  7         "   ERASE
LEFT   BYTE  8         "   LEFT
RIGHT  BYTE  9         "   RIGHT
ENTER  BYTE  13        "   ENTER
CRSR   BYTE  126       CURSOR CHARACTER
*
       EVEN
FAC    EQU   >834A    FPOINT AKKUMULATOR
STATUS EQU   >837C    GPL STATUS BYTE
KEYBRD EQU   >8375    ASCII WERT DER GEDRUECKTEN TASTE
JOYY   EQU   >8376    JOYSTICK WERTE
JOYX   EQU   >8377
GPLWS  EQU   >83E0    GPL ARBEITSREGISTER
*
ERRBA  EQU   >1600    BAD ARGUMENT
CFI    EQU   >1200    REAL-INTEGER UEBERSETZUNG
CIF    EQU   >2300    INTEGER-FPOINT UEBERSETZUNG
*
```

```
CLRDTA DATA >0000,>0000,>0000,>0000   32 LEERZEICHEN
       DATA >0000,>0000,>0000,>0000
       DATA >0000,>0000,>0000,>0000
       DATA >0000,>0000,>0000,>0000
SPTCHR DATA >0000,>1010,>7C10,>1000   SPRITE CHARACTER  ("+")
SPTDTA DATA >2020,>0B0F,>D000         WERTE FUER SPRITE AUFRUF
*
       COPY "DSK1.@SUBPROG#"          COPY DIREKTIVEN FUER ALLE
       COPY "DSK1.@DRAW#"             ANDEREN PROGRAMMTEILE
       COPY "DSK1.@CHARSCAN#"
       COPY "DSK1.@INVERT#"
       COPY "DSK1.@DSPACPT#"
       COPY "DSK1.@SCROLL#"
       COPY "DSK1.@SCRMEMO#"
       COPY "DSK1.@SELECT#"
*
       END
```

```
*PROGRAMMSEGMENT "@INVERT#"
*INVERTIERT DAS ZEICHENFELD
*LAENGE=42 ZEILEN
*
INVERT LWPI MYWS              EIGENE ARBEITSREGISTER LADEN
*
       CLR  R3                R3=HILFSREGISTER FUER SPALTENPOSITION
       CLR  R4                R4=HILFSREGISTER FUER ZEILENPOSITION
       LI   RØ,>ØØ22          ANFANGSADRESSE IM VDP-RAM
       CLR  R1                BYTE LESEN
INVLP1 BLWP @VSBR
       NOP
       C    R1,@CHAR8Ø        MSB R1=8Ø
       JEQ  CHRSTØ            DANN  ASCII 32 > < GEGEN
*                             ASCII 126 >~< TAUSCHEN
       BL   @CHRST1           MSB R1=DE
*                             ASCII 126>~< GEGEN
*                             ASCII 32 > < TAUSCHEN
       JMP  NXTLP1
CHRSTØ BL   @CHRST2
NXTLP1 INC  RØ                NAECHSTE SPALTE
       INC  R3                HILFSREGISTER FUER SPALTENPOSITION
       CI   R3,>ØØ1Ø          16 SPALTEN GEPRUEFT?
       JNE  INVLP1
       AI   RØ,>ØØ1Ø          NAECHSTE ZEILE
       CLR  R3                HILFSREGISTER LOESCHEN
       INC  R4                HILFSREGISTER ZEILE ERHOEHEN
       CI   R4,>ØØ1Ø          ALLE ZEILEN BEARBEITET?
       JNE  INVLP1            WENN NICHT-WEITER
       BL   @BSRTN            ZURUECK ZU BASIC
*
CHRST1
       LI   R1,>8ØØØ          LEERZEICHEN SETZEN
       BLWP @VSBW
       NOP
       RT
*
CHRST2
       LI   R1,>DEØØ          TILDE >~< SETZEN
       BLWP @VSBW
       NOP
       RT
```

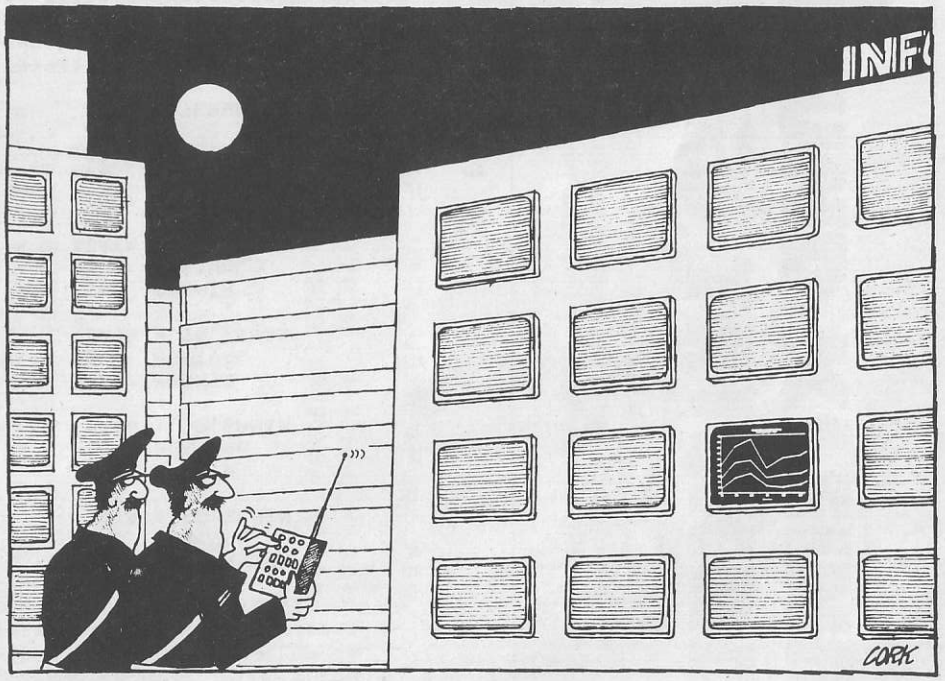
```
*PROGRAMMSEGMENT "@SCROLL#"
*LAENGE=3ØØ ZEILEN
*
SCROLL LWPI MYWS              EIGENE REGISTER LADEN
*
       LI   R1,5              SCROLLRICHTUNG IST 5.LINK PARAMETER
       MOV  R1,@PRMTR
       BL   @GETVAL           WERT GEWINNEN
       DATA 1,4,SCRDIR        UND IN SCRDIR SPEICHERN
*
       CLR  @FLAGØ1           FLAGØ1 LOESCHEN
       INC  @FLAGØ1           FLAGØ1 =1
       C    @SCRDIR,@FLAGØ1   RICHTUNG=1?
       JNE  SCR2              NEIN--> SCR2
       BL   @SCRVL1           SCROLLPARM. GEWINNEN
       BL   @SCDN             UPRG. SCROLLDOWN
       BL   @BSRTN            BASIC RETURN
*
SCR2   INC  @FLAGØ1           FLAGØ1=2
       C    @SCRDIR,@FLAGØ1   RICHTUNG=2?
       JNE  SCR3              NEIN--> SCR3
       BL   @SCRVL1           SCROLLPARM.GEWINNEN
       BL   @SCUP             UPRG.SCROLLUP
       BL   @BSRTN            BASIC RETURN
*
SCR3   INC  @FLAGØ1           SCROLLRICHTUNG 3 =RECHTS
       C    @SCRDIR,@FLAGØ1
       JNE  SCR4
       BL   @SCRVL2
       BL   @SCRT
       BL   @BSRTN
*
SCR4   BL   @SCRVL2           SCROLLRICHTUNG 4 =LINKS
       BL   @SCLT
       BL   @BSRTN
*
SCDN   MOV  R11,@RTLEV3       RETURNADRESSE GEWINNEN
       LIMI Ø
       BL   @SCVLC1           SCROLL WERTE KONTROLLE
*
       CLR  @FLAGØ2
       MOV  @SPMIN,@SPPOS
*
DNLP1  CLR  @FLAGØ1
       MOV  @ZEMAX,@ZEPOS
       DEC  @ZEPOS
DNLP2  BL   @VALCLC           ERSTE POSITION IN VDP-RAM
       MOV  @NEWPOS,RØ        BERECHNEN
       LI   R1,SCRBUF
       MOV  @SPDIF,R2         SPMAX-SPMIN ERGIBT SPDIF
       BLWP @VMBR             EINE ZEILE LESEN
       INC  @ZEPOS            UND DIE GELESENE ZEILE
       BL   @VALCLC           EINE ZEILE TIEFER
       MOV  @NEWPOS,RØ        WIEDER IN DAS VDP-RAM
       LI   R1,SCRBUF         SCHREIBEN.
       MOV  @SPDIF,R2
       BLWP @VMBW
       DECT @ZEPOS            NAECHSTE ZEILE LESEN
       INC  @FLAGØ1           ZEMAX-ZEMIN=ZEDIF
```

```
       C    @FLAGØ1,@ZEDIF  ALLE ZEILEN BEARBEITET?
       JNE  DNLP2           WENN NICHT DANN WEITER
*
       MOV  @ZEMIN,@ZEPOS   LETZTE ZEILE
       BL   @VALCLC         BERECHNEN
       MOV  @NEWPOS,RØ      UND DANN MIT LEER-
       LI   R1,CLRDTA       ZEICHEN AUS CLRDTA
       MOV  @SPDIF,R2       UEBERSCHREIBEN
       BLWP @VMBW
       INC  @FLAGØ2
       C    @FLAGØ2,@SCRCNT SCROLLVORGANG ENTSPRECHEND
       JNE  DNLP1           OFT WIE VORGEGEBEN WIEDERHOLT?
*
       MOV  @RTLEV3,R11     RETURNADRESSE GEWINNEN UND
       LIMI 2
       RT                   ZURUECK
*****************************************
*SCUP BEINHALTET DAS GLEICHE PROGRAMM     *
*WIE UNTER SCDN BESCHRIEBEN.              *
*DER EINZIGE UNTERSCHIED LIEGT DARIN      *
*DAS DIE ZEILEN NACH OBEN GESCHOBEN WERDEN*
*****************************************
SCUP   MOV  R11,@RTLEV3
       LIMI Ø
       BL   @SCVLC1
*
       CLR  @FLAGØ2
       MOV  @SPMIN,@SPPOS
*
UPLP1  CLR  @FLAGØ1
       MOV  @ZEMIN,@ZEPOS
       INC  @ZEPOS
UPLP2  BL   @VALCLC
       MOV  @NEWPOS,RØ
       LI   R1,SCRBUF
       MOV  @SPDIF,R2
       BLWP @VMBR
       DEC  @ZEPOS
       BL   @VALCLC
       MOV  @NEWPOS,RØ
       LI   R1,SCRBUF
       MOV  @SPDIF,R2
       BLWP @VMBW
       INCT @ZEPOS
       INC  @FLAGØ1
       C    @FLAGØ1,@ZEDIF
       JNE  UPLP2
*
       MOV  @ZEMAX,@ZEPOS
       BL   @VALCLC
       MOV  @NEWPOS,RØ
       LI   R1,CLRDTA
       MOV  @SPDIF,R2
       BLWP @VMBW
       INC  @FLAGØ2
       C    @FLAGØ2,@SCRCNT
       JNE  UPLP1
*
       MOV  @RTLEV3,R11
       LIMI 2
       RT
*
SCRT   MOV  R11,@RTLEV3     RETURNADRESSE GEWINNEN
       LIMI Ø
       BL   @SCVLC2

       CLR  @FLAGØ2
*
RTLP1  CLR  @FLAGØ1
       MOV  @ZEMIN,@ZEPOS   ANFANGSPOSITION
       MOV  @SPMIN,@SPPOS   IM VDP-RAM
RTLP2  BL   @VALCLC         BERECHNEN
       MOV  @NEWPOS,RØ      EINE ZEILE
       LI   R1,SCRBUF       NACH SCRBUF UEBERTRAGEN
       MOV  @SPDIF,R2       LAENGE DER ZEILE WIEDER
       BLWP @VMBR           SPMAX-SPMIN=SPDIF
       LI   R1,>8ØØØ        LEERZEICHEN AN DIE
       BLWP @VSBW           ANFANGSADRESSE SCHREIBEN
       INC  RØ              SPALTE=SPALTE+1
       LI   R1,SCRBUF       SCRBUF WIEDER IN DAS
       MOV  @SPDIF,R2       VDP-RAM SCHREIBEN
       BLWP @VMBW
       INC  @ZEPOS          NAECHSTE ZEILE
       INC  @FLAGØ1
       C    @FLAGØ1,@ZEDIF  ALLE ZEILEN VERARBEITET?
       JNE  RTLP2           NEIN-->WEITER
       INC  @FLAGØ2
       C    @FLAGØ2,@SCRCNT SCROLLCOUNT ERREICHT?
       JNE  RTLP1           NEIN-->WEITER
*
       MOV  @RTLEV3,R11     RETURNADRESSE GEWINNEN
       LIMI 2
       RT                   UND ZURUECK
*****************************************
*SCLT BEINHALTET DAS GLEICHE PROGRAMM     *
*WIE IN SCRT BESCHRIEBEN.AUCH HIER LIEGT DER*
*UNTERSCHIED NUR IN DER RICHTUNG IN       *
*DER DIE ZEILEN GESCHOBEN WERDEN.         *
*****************************************
SCLT   MOV  R11,@RTLEV3
       LIMI Ø
       BL   @SCVLC2
*
       CLR  @FLAGØ2
*
LTLP1  CLR  @FLAGØ1
       MOV  @ZEMIN,@ZEPOS
       MOV  @SPMIN,@SPPOS
       INC  @SPPOS
LTLP2  BL   @VALCLC
       MOV  @NEWPOS,RØ
       LI   R1,SCRBUF
       MOV  @SPDIF,R2
       BLWP @VMBR
       DEC  @SPPOS
       BL   @VALCLC
       MOV  @NEWPOS,RØ
       LI   R1,SCRBUF
```

```
       MOV  @SPDIF,R2
       BLWP @VMBW
       MOV  @SPMAX,@SPPOS
       BL   @VALCLC
       MOV  @NEWPOS,RØ
       LI   R1,>8ØØØ
       BLWP @VSBW
       MOV  @SPMIN,@SPPOS
       INC  @SPPOS
       INC  @ZEPOS
       INC  @FLAGØ1
       C    @FLAGØ1,@ZEDIF
       JNE  LTLP2
       INC  @FLAGØ2
       C    @FLAGØ2,@SCRCNT
       JNE  LTLP1
*
       MOV  @RTLEV3,R11
       LIMI 2
       RT
*
SCVLC1 C    @ZEMAX,@ZEMIN   ZEMAX >ZEMIN?
       JGT  CON1            JA
       B    @BADARG         NEIN -->"BAD ARGUMENT " UND STOP
CON1   C    @SPMAX,@SPMIN   SPMAX GROESSER ODER GLEICH SPMIN
       JGT  CON2
       JEQ  CON2
       B    @BADARG         WENN NICHT DANN "BAD ARGUMENT" UND STOP
*
CON2   MOV  @SPMAX,@SPDIF   SPALTENDIFFERENZ
       S    @SPMIN,@SPDIF   BILDEN
       INC  @SPDIF
*
       MOV  @ZEMAX,@ZEDIF   ZEILENDIFFERENZ
       S    @ZEMIN,@ZEDIF   BILDEN
       INCT @ZEDIF
       C    @ZEDIF,@SCRCNT  SCROLLCOUNT AN
       JGT  VLCRT1          ZEILENDIFFERENZ
       MOV  @ZEDIF,@SCRCNT  ANPASSEN
       DEC  @SCRCNT
*
VLCRT1 DECT @ZEDIF
       RT
*
SCVLC2 C    @ZEMAX,@ZEMIN   ZEMAX GROESSER ODER GLEICH ZEMIN
       JGT  CON11
       JEQ  CON11
       B    @BADARG         "BAD ARGUMENT" UND STOP
CON11  C    @SPMAX,@SPMIN   SPMAX GROESSER SPMIN
       JGT  CON22
       B    @BADARG         "BAD ARGUMENT"UND STOP
*
CON22  MOV  @ZEMAX,@ZEDIF   ZEILENDIFFERENZ BILDEN
       S    @ZEMIN,@ZEDIF
       INC  @ZEDIF
*
       MOV  @SPMAX,@SPDIF   SPALTENDIFFERENZ BILDEN
       S    @SPMIN,@SPDIF
       INCT @SPDIF
   C    @SPDIF,@SCRCNT  SCROLLCOUNT  AN
   JGT  VLCRT2          SPDIFFERENZ ANPASSEN
   MOV  @SPDIF,@SCRCNT
   DEC  @SCRCNT
*
VLCRT2 DECT @SPDIF
       RT
****************************
*PARAMETER-UEBERNAHME FUER*
*SCROLLDOWN UND SCROLLUP  *
****************************
SCRVL1 MOV  R11,@RTLEV2
       LI   R1,1
       MOV  R1,@PRMTR
       BL   @GETVAL
       DATA 1,23,ZEMIN
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,32,SPMIN
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 2,24,ZEMAX
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,32,SPMAX
*
       INCT @PRMTR
       BL   @GETVAL
       DATA 1,24,SCRCNT
*
       MOV  @RTLEV2,R11
       RT
****************************
*PARAMETER-UEBERNAHME FUER *
*SCROLLRIGHT UND SCROLLLEFT*
****************************
SCRVL2 MOV  R11,@RTLEV2
       LI   R1,1
       MOV  R1,@PRMTR
       BL   @GETVAL
       DATA 1,24,ZEMIN
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,32,SPMIN
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,24,ZEMAX
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,32,SPMAX
*
       INCT @PRMTR
       BL   @GETVAL
```

```
*DAS PROGRAMMSEGMENT "@SELECT#"
*ENTHAELT DIE ZEICHENROUTINE
*LAENGE=73 ZEILEN
*
SELECT LWPI MYWS
*
       BL   @SLCTVL          PARAMETER HOLEN
       BL   @SPTSET          "CALL SPRITE...."
       BL   @VALTST          ZEICHENPARAMETER PRUEFEN
       BL   @VALCLC          POSITION BERECHNEN
       BL   @SPTLCT          "CALL LOCATE..."
*
SELCT1 BL    @DELAY          PAUSE
       LI    R0,>0100        KEYBOARD LINKE SEITE&JOYST.1
       MOVB  R0,@>8374
       BLWP  @KSCAN          TASTATUR ABFRAGEN
       C     @JOYY,@H00      JOYSTICK NICHT BEWEGT
       JNE   SELCT3
*
       CB    @KEYBRD,@KEY18 FEUERTASTE GEDRUECKT
       JEQ   SELCT2          JA,DANN ZURUECK
       JNE   SELCT1          NEIN,WIEDER VON VORNE
*
SELCT2 BL    @RTVAL1         ZEILE UND SPALTE AN LINK
       BL    @BSRTN          UEBERGEBEN UND ZURUECK ZUM BASIC
*
SELCT3 BL    @JOYCMP         JOYSTICK AUSWERTEN
       BL    @VALTST         WERTE TESTEN
       BL    @VALCLC         POSITION BERECHNEN
       BL    @SPTLCT         "CALL LOCATE..."
       BL    @DELAY          PAUSE
       JMP   SELCT1
****************************
*SLCTVAL UEBERNIMMT DIE    *
*PARAMETER FUER DAS SELECT *
*PROGRAMM.                 *
****************************
SLCTVL MOV  R11,@RTLEV2
       LI   R1,1
       MOV  R1,@PRMTR
       BL   @GETVAL
       DATA 1,24,ZEMIN
       DEC  @ZEMIN
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,32,SPMIN
       DEC  @SPMIN
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,24,ZEMAX
       INC  @ZEMAX
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,32,SPMAX
       INC  @SPMAX
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,24,ZEPOS
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 1,32,SPPOS
*
       INC  @PRMTR
       BL   @GETVAL
       DATA 10,2000,DLYTME
*
       MOV  @RTLEV2,R11
       RT
```



```
*PROGRAMMSEGMENT "@SUBPROG#"
*ENTHAELT ALLE MEHRFACH
*GENUTZTEN UNTERPROGRAMME
*LAENGE=130 ZEILEN
*
DELAY  MOV  @DLYTME,15    HOLT WERT FUER DELAY NACH R15
DELYLP NOP                TUT NICHTS
       DEC  15            R15=R15-1
       MOV  15,15         R15=0?
       JNE  DELYLP        WENN NICHT,DANN WEITERZAEHLEN
       RT
*****************************************
*VALCLC ERRRECHNET AUF DER BASIS        *
*VON ZEILE(ZEPOS)&SPALTE(SPPOS) NACH DER*
*FORMEL (ZEILE-1*32)+SPALTE-1 DIE       *
*NEUE POSITION IM VDP-RAM               *
*****************************************
VALCLC MOV  @ZEPOS,R0   ZEPOS NACH R0
       DEC  R0          ZEPOS=ZEPOS-1
       SLA  R0,5        *32
       A    @SPPOS,R0   +SPPOS
       DEC  R0          -1
       MOV  R0,@NEWPOS  NEUE POSITION NACH NEWPOS(ition
       RT
*****************************************************
*GETVAL  UBERNIMMT PARAMETER UND UEBERPRUEFT         *
*OB SIE MIT DEN ANGEGEBENEN GRENZEN UEBEREINSTIMMEN  *
*****************************************************
*
GETVAL MOV  *R11+,R2      UNTERE GRENZE
       MOV  *R11+,R3      OBERE  GRENZE
       MOV  *R11+,R4      ZIELADRESSE
       MOV  R11,@RTLEV1   RETURNADRESSE SICHERN
*
       DEC  R2            GRENZEN FUER VERGLEICH
       INC  R3            AUFBEREITEN
*
       CLR  R0            LINK  PARAMETER NUMMER
       MOV  @PRMTR,R1     HOLEN
       BL   @GETPAR       WERT HOLEN UND IN
       MOV  @FAC,*R4      ZIEL SPEICHERN
*
       C    *R4,R2        UNTERE GRENZE TESTEN
       JGT  TSTOG         OK.
       B    @BADARG       GRENZE VERLETZT
TSTOG  C    *R4,R3        OBERE GRENZE TESTEN
       JLT  GPRT          OK.
       B    @BADARG       GRENZE VERLETZT

GPRT   MOV  @RTLEV1,R11   RETURNADRESSE WIEDERGEWINNEN
       RT                 UND AB DIE POST ZURUECK
*
GETPAR BLWP @NUMREF       NUMERISCHEN PARAMETER
       BLWP @XMLLNK       UEBERNEHMEN UND IN INTEGER
       DATA CFI           TRANSFORMIEREN
       RT                 ZURUECK
*
BADARG LI   R0,ERRBA      FEHLERCODE NACH HBYTE R0
       BLWP @ERR          ANZEIGE "BAD ARGUMENT" UND STOP
*
BSRTN  MOV  @H00,@STATUS  STATUS LOESCHEN
       LWPI GPLWS         GPL ARBEITSREGISTER LADEN
       B    @>0070        ZURUECK ZUM BASIC
*********************************
*JOYCoMPare VERGLEICHT DIE WERTE *
*FUER JOYY UND JOYX UND ERMITTELT*
*DIE JEWEILIGE RICHTUNG.        *
*********************************
JOYCMP C    @JOYY,@JYUP    JOYSTICK NACH OBEN?
       JEQ  MVUP           JA!
       C    @JOYY,@JYRT    JOYSTICK NACH RECHTS?
       JEQ  MVRT           JA!
       C    @JOYY,@JYDN    UND IMMER SO WEITER...
       JEQ  MVDN
       C    @JOYY,@JYLT
       JEQ  MVLT
       C    @JOYY,@JYLTUP
       JEQ  MVLTUP
       C    @JOYY,@JYLTDN
       JEQ  MVLTDN
       C    @JOYY,@JYRTUP
       JEQ  MVRTUP
       C    @JOYY,@JYRTDN
       JEQ  MVRTDN
       JMP  JOYRTN
************************
MVUP   DEC  @ZEPOS        NACH OBEN--> ZEILE=ZEILE-1
       JMP  JOYRTN        ZURUECK
*
MVDN   INC  @ZEPOS        NACH UNTEN-->ZEILE=ZEILE+1
       JMP  JOYRTN        ZURUECK
*
MVRT   INC  @SPPOS        NACH RECHTS-->SPALTE=SPALTE+1
       JMP  JOYRTN        ZURUECK
*
MVLT   DEC  @SPPOS        NACH LINKS-->SPALTE=SPALTE-1
       JMP  JOYRTN
*
MVRTUP INC  @SPPOS        NACH RECHTS OBEN-->SPALTE=SPALTE+1
       DEC  @ZEPOS        ZEILE=ZEILE-1
       JMP  JOYRTN        ZURUECK
*
MVRTDN INC  @SPPOS        NACH RECHTS UNTEN-->SPALTE=SPALTE+1
       INC  @ZEPOS        ZEILE=ZEILE+1
       JMP  JOYRTN        ZURUECK
*
MVLTUP DEC  @ZEPOS        NACH LINKS OBEN-->ZEILE=ZEILE-1
       DEC  @SPPOS        SPALTE=SPALTE-1
       JMP  JOYRTN        ZURUECK
*
MVLTDN INC  @ZEPOS        NACH LINKS UNTEN-->ZEILE=ZEILE+1
       DEC  @SPPOS        SPALTE=SPALTE-1
*
JOYRTN RT                 ZURUECK
*
RTVAL1 LI   R1,8          ZEILE UND SPALTE SIND DER
       JMP  RTVAL3        8.UND DER 9. PARAMETER IM LINKAUFRUF
RTVAL2 LI   R1,9          ZEILE UND SPALTE SIND DER 9.UND DER 10.
```

```
*                       PARAMETER IM LINKAUFRUF
RTVAL3 CLR  RØ
MOV  @ZEPOS,@FAC    ZEILE NACH FPOINTAKKU.
BLWP @XMLLNK        UMWANDLUNG INTEGER-->FPOINT
DATA CIF
BLWP @NUMASG        PARAMETER UEBERGEBEN
INC  R1             NAECHSTER PARAMETER IST SPALTE
MOV  @SPPOS,@FAC    SPALTE NACH FPOINTAKKU.
BLWP @XMLLNK        UMWANDLUNG INTEGER-->FPOINT
DATA CIF
BLWP @NUMASG        PARAMETER UEBERGEBEN
RT                  ZURUECK
```

```
*PROGRMMSEGMENT "@SCRMEMO#
*LAENGE=15 ZEILEN
*
SCRSAV LWPI MYWS            EIGENE ARBEITSREGISTER LADEN
       LI   RØ,>ØØØØ        STARTADRESSE FUER SCREEN IMAGE TABLE
       LI   R1,SCREEN       STARTADRESSE  FUER SCREENBUFFER
       LI   R2,768          768 CHARACTERS WERDEN GELESEN
       BLWP @VMBR
       BL   @BSRTN          --->BASIC
*
SCRRET LI   RØ,>ØØØØ        STARTADRESSE SCREEN IMAGE TABLE
       LI   R1,SCREEN       STARTADRESSE SCREENBUFFER
       LI   R2,768          768 BYTES WIEDER
       BLWP @VMBW           IN DAS VDP-RAM SCHREIBEN
       BL   @BSRTN          --->BASIC
```

```
*PROGRAMMSEGMENT "@CHARSCAN#"
*TASTET DAS ZEICHENFELD AB UND
*LIEFERT DIE STRING VARIABLE
*CHAR# AN DAS BASIC PROGRAMM
*LAENGE=1Ø2 ZEILEN
*
CHRSCN LWPI MYWS            EIGENE ARBEITSREGISTER
*
       LI   RØ,>4ØØØ        ERSTES BYTE IN STRING ENT-
       MOVB RØ,@STRING      HAELT DIE LAENGE.
       LI   R6,STRING+1     R6 IST DER ZEIGER AUF STRING
       CLR  R7              R7 WIRD ALS HILFSREGISTER BENUTZT
       LI   RØ,>ØØ22        ANFANGSADRESSE DES ZEICHENFELDES
       LI   R1,CHRBUF       ADRESSE DES CHARACTERBUFFERS
       LI   R2,4            4 BYTE NACH CHRBUF UEBERTRAGEN
**********************************
*SCNLP1 TASTET DIE LINKE SEITE DES  *
*ZEICHENFELDES ZEILE FUER ZEILE AB  *
**********************************
SCNLP1 INC  R7
       BLWP @VMBR
       BL   @SUBSCN
       MOVB @CHRBYT,*R6+    CHRBYT IN STRING SCHREIBEN
       AI   RØ,>ØØØ4        WIEDER VIER BYTE LESEN
       LI   R1,CHRBUF
       BLWP @VMBR
       BL   @SUBSCN         ZU CHRBUF PASSENDES CHRBYT SUCHEN
       MOVB @CHRBYT,*R6+    CHRBYT IN STRING SCHREIBEN
       AI   RØ,>ØØ1C        NAECHSTE ZEILE
       LI   R1,CHRBUF
       CI   R7,>ØØ1Ø        ALLE ZEILEN GELESEN?
       JNE  SCNLP1          WENN NICHT DANN WEITER
*
       CLR  R7
       LI   RØ,>ØØ2A        VIER BYTE IN
       LI   R1,CHRBUF       CHARBUF EINLESEN
       LI   R2,4
**********************************
*SCNLP2 TASTET DIE RECHTE HAELFTE*
*DES ZEICHENFELDES AB               *
**********************************
SCNLP2 INC  R7
       BLWP @VMBR
       BL   @SUBSCN         VERGLEICHEN MIT HXDATA
       MOVB @CHRBYT,*R6+    CHARBYT IN STRING SCHREIBEN
       AI   RØ,>ØØØ4        NEUE ADRESSE
       LI   R1,CHRBUF       WIEDER VIER BYTE
       BLWP @VMBR           IN CHRBUF EINLESEN
       BL   @SUBSCN         VERGLEICHEN MIT HXDATA
       MOVB @CHRBYT,*R6+    CHRBYT IN STRING SCHREIBEN
       AI   RØ,>ØØ1C        NAECHSTE ZEILE
       LI   R1,CHRBUF
       CI   R7,>ØØ1Ø        ALLE 16 ZEILEN ABGETASTET?
       JNE  SCNLP2          WENN NICHT  DANN WEITER
*
*
       MOVB @HØØ,@STATUS
       CLR  RØ
       LI   R1,1
```

```
       LI   R2,STRING
       BLWP @STRASG
       BL   @BSRTN
***********************************************
*SUBSCAN VERGLEICHT DIE VIER BYTES IN CHRBUF WORT*
*FUER WORT MIT HXDATA.JE EIN HXBYTE ENTSPRICHT   *
*EINEM WORT AUS HEXDTA.                          *
***********************************************
SUBSCN CLR  @CHRBYT          CHRBYT LOESCHEN
       CLR  @FLAGØ1          FLAGØ1  LOESCHEN
       LI   R3,HEXDTA        STARTADRESSE HEXDTA
       LI   R4,CHRBUF        STARTADRESSE CHRBUF
       LI   R5,HXBYTE        STARTADRESSE HXBYTE
       MOVB *R5+,@CHRBYT     ERSTES BYTE AUS HXBYTE
*                            NACH CHRBYT
COMP1  INC  @FLAGØ1          FLAGØ1 =1
       C    *R4+,*R3+        ERSTES WORT CHRBUF=ERSTES
*                            WORT IN HEXDTA?
*
       JNE  NXTWRD           WENN NICHT DANN NAECHSTES WORT
       C    @FLAGØ1,@VALUE2  ZWEITES WORT GEFUNDEN?
       JEQ  SCANRT           JA->ZURUECK
       JMP  COMP1            ZWEITES WORT VERGLEICHEN
*
NXTWRD
       C    @FLAGØ1,@VALUE1  FLAGØ1=1? EIN WORT GEFUNDEN
       JEQ  NXWRD1
*
       C    @FLAGØ1,@VALUE2  FLAGØ1=2? BEIDE WORTE GEFUNDEN
       JEQ  NXWRD2
*
NXWRD1 MOV  *R3+,@WRDBUF     NAECHSTES WORT AUS HXDATA
       LI   R4,CHRBUF        R4  WIEDER AUF ANFANG VON CHRBUF SETZEN
       MOVB *R5+,@CHRBYT     NAECHSTES HXBYTE NACH CHRBYT
       CLR  @FLAGØ1          FLAGØ1 LOESCHEN
       JMP  COMP1
*
NXWRD2 LI   R4,CHRBUF        R4 WIEDER AUF ANFANG VON CHRBUF SETZEN
       CLR  @FLAGØ1          FLAGØ1 LOESCHEN
       MOVB *R5+,@CHRBYT     NAECHSTES HXBYTE NACH CHRBYT
       JMP  COMP1            WIEDER VERGLEICHEN
*
SCANRT RT                    ZURUECK
```

```
*DAS PROGRAMMSEGMENT "@DRAW#"
*ENTHAELT DIE ZEICHENROUTINE
*LAENGE=152 ZEILEN
*
DRAW   LWPI MYWS             EIGENE ARBEITSREGISTER
*
       CLR  @TACT            TACT AUF NULL SETZEN
       BL   @DRWVAL          LINKPARAMETER FUER ZEICHENPROGRAMM UEBERNEHMEN
       BL   @SPTSET          SPRITE CURSOR SETZEN >+<
       BL   @VALTST          UEBERPRUEFEN OB ZEPOS UND SPPOS INNERHALB DER
*                            ZULAESSIGEN GRENZEN SIND
       BL   @VALCLC          NEUE POSITION IN VDP-RAM ERRECHNEN
       BL   @SPTLCT          SPRITE CURSOR AUF DIESE POSITION SETZEN
*************************************************************
*DRAWL1 UBERPRUEFT OB TACT=TVRG,WENN JA DANN                 *
*WERDEN ZEILE UND SPALTE AN DAS BASICPROGRAMM UEBERGEBEN#
*************************************************************
DRAWL1 BL   @DELAY           PAUSE EINSCHALTEN
       INC  @TACT            TACT UM EINS ERHOEHEN
       C    @TACT,@TVRG      TACT MIT TVRG VERGLEICHEN
       JNE  DRAWL3           WENN NOCH NICHT GLEICH DANN WEITER MIT DRAWL3
       BL   @RTVAL2          ZEILE UND SPALTE AN DIE LINKPARAMETER UEBERGEBEN
       BL   @BSRTN           ZURUECK INS BASIC
*
DRAWL2 CLR  @TACT            TACT WIEDER AUF NULL
       BL   @DELAY           PAUSE
DRAWL3 LI   RØ,>ØlØØ         LINKE TASTATURSEITE UND DER
       MOVB RØ,@>8374        JOYSTICK 1 WIRD GEPRUEFT.
       BLWP @KSCAN
       C    @JOYY,@HØØ       JOYSTICK BEWEGT ?
       JEQ  DRAWL4           WENN NICHT DANN DRAWL4
*
       BL   @JOYCMP          WOHIN WURDE DER JOYSTICK BEWEGT?
       BL   @VALTST          NOCH IN DEN GEWUENSCHTEN GRENZEN?
       BL   @VALCLC          NEUE POSITION ERRRECHNEN
       BL   @CHRSET          AN DER NEUEN POSITION ENTWEDER EIN
*                            ZEICHEN SETZEN ODER LOESCHEN.
       BL   @SPTLCT          CURSOR >+< AUF NEUE POSITION
       BL   @DELAY           PAUSE
       JMP  DRAWL2           UND WEITER
*
DRAWL4 CB   @KEYBRD,@KEY18   FEUERTASTE GEDRUECKT?
       JNE  DRAWL1           NEIN
       BL   @CHRSET          SONST WIEDER ZEICHEN SETZEN
       JMP  DRAWL2           UND WEITER
*****************************************
*VALTST UEBERPRUEFT,OB DIE NEUE ZEILEN UND  *
*SPALTENPOSITION MIT DEN ANGEGEBENEN GRENZEN#
*UEBEREINSTIMMMT,UND KORRIGIERT DIE WERTE   *
*ENTSPRECHEND.                              *
*****************************************
VALTST C    @ZEPOS,@ZEMIN    UNTERE ZEILENGRENZE ERREICHT?
       JEQ  ZINC             WENN JA DANN KORRIGIEREN
       JMP  ZEMAXC           NEIN,DANN OBERE GRENZE PRUEFEN
ZINC   INC  @ZEPOS           ZEILE =ZEILE+1
       JMP  SPMINC           UNTERE SPALTENGRENZE PRUEFEN
*
ZEMAXC C    @ZEPOS,@ZEMAX    OBERE ZEILENGRENZE ERREICHT?
       JEQ  ZDEC             WENN JA DANN KORRIGIEREN
```

```
       JMP   SPMINC
ZDEC   DEC   @ZEPOS          ZEILE=ZEILE-1
*
SPMINC C     @SPPOS,@SPMIN   UNTERE SPALTENGRENZE ERREICHT?
       JEQ   SINC            WENN JA ,DANN KORRIGIEREN
       JMP   SPMAXC          OBERE GRENZE PRUEFEN
SINC   INC   @SPPOS          SPALTE =SPALTE+1
       JMP   VTSTRT          ZURUECK
*
SPMAXC C     @SPPOS,@SPMAX   OBERE SPALTENGRENZE ERREICHT
       JEQ   SDEC            WENN JA DANN KORREKTUR
       JMP   VTSTRT          ZURUECK
SDEC   DEC   @SPPOS          SPALTE=SPALTE-1
VTSTRT RT                    ZURUECK
*************************
*
CHRSET LIMI Ø
       CB    @KEYBRD,@KEY18 FEUERTASTE GEDRUECKT?
       JEQ   NEWCHR         WENN JA DANN CHARACTER SETZEN
*
DELCHR MOV   @NEWPOS,RØ     NEUE POSITION NACH RØ
       LI    R1,>8ØØØ       LEERZEICHEN AUF NEUE POSITION
       BLWP  @VSBW
       JMP   CSETRT         ZURUECK
*
NEWCHR MOV   @NEWPOS,RØ     NEUE POSITION NACH RØ
       LI    R1,>DEØØ       TILDE >~< AUF NEUE POSITION SETZEN
       BLWP  @VSBW
CSETRT RT                    ZURUECK
************************
*
SPTSET LI    RØ,>Ø3ØØ       SPRITE SETZEN,ENTSPRICHT  DEM CALL
       LI    R1,SPTDTA      SPRITE BEFEHL DES EXTENDED BASIC
       LI    R2,6
       BLWP  @VMBW
       RT
***********************************
*SPTLCT SETZT DEN SPRITE CURSOR AUF*
*DIE POSITION ZEPOS*8 UND SPPOS*8  *
***********************************
SPTLCT LIMI Ø
       MOV   @ZEPOS,R1      ZEPOS NACH R1
       DEC   R1             ZEPOS=ZEPOS-1
       LI    RØ,>Ø3ØØ       BEGINN DER SPRITE ATTRIBUTE LIST
       SLA   R1,11          ZEPOS*8
       BLWP  @VSBW          IN LISTE EINTRAGEN
       MOV   @SPPOS,R1      SPPOS NACH R1
       DEC   R1             SPPOS=SPPOS-1
       INC   RØ             NAECHSTE EINTRAGSTELLE IM SPR.ATTR.TBL.
       SLA   R1,11          SPPOS*8
       BLWP  @VSBW
       RT                   ZURUECK
*************************
DRWVAL MOV   R11,@RTLEV2     RETURNADRESSE SICHERN
       LI    R1,1           ZEMIN UEBERNEHMEN
       MOV   R1,@PRMTR      LINK PARAMETER 1 FUER ZEMIN
       BL    @GETVAL
       DATA  1,24,ZEMIN     GRENZEN 1--24
       DEC   @ZEMIN         GRENZE AUFBEREITEN
*
       INC   @PRMTR
       BL    @GETVAL
       DATA  1,32,SPMIN
       DEC   @SPMIN
*
       INC   @PRMTR
       BL    @GETVAL
       DATA  1,24,ZEMAX
       INC   @ZEMAX
*
       INC   @PRMTR
       BL    @GETVAL
       DATA  1,32,SPMAX
       INC   @SPMAX
*
       INC   @PRMTR
       BL    @GETVAL
       DATA  1,24,ZEPOS
*
       INC   @PRMTR
       BL    @GETVAL
       DATA  1,32,SPPOS
*
       INC   @PRMTR         UEBERNAHME VON DLYTME
       BL    @GETVAL        DLYTME BESTIMMT DIE LAUF-
       DATA  1Ø,2ØØØ,DLYTME GESCHWINDIGKEIT DES CURSORS
*
       INC   @PRMTR         UEBERNAHME DER VORGABEZEIT
       BL    @GETVAL        BEI DER EIN RUECKSPRUNG
       DATA  1Ø,1ØØØ,TVRG   IN DAS BASIC ERFOLGT
*
       MOV   @RTLEV2,R11    RETURNADRESSE WIEDERGEWINNEN
       RT                   ZURUECK
       DATA  1,33,SCRCNT
*
       MOV   @RTLEV2,R11
       RT
*
```

```
*PROGRAMMSEGMENT "@DSPACPT#"
*LAENGE=190 ZEILEN
*
DSPLAY LWPI MYWS              EIGENE ARBEITSREGISTER LADEN
       BL   @GETRCS           ROW,COL,SIZE UEBERNEHMEN
*
       CLR  R0
       LI   R1,4              STRING IST DER VIERTE LINKPARAMETER
       LI   R2,DASTRG
       MOVB @BYT28,*R2        STRINGLAENGE=28 BYTES
       BLWP @STRREF           STRING UEBERNEHMEN
       MOVB @DASTRG,LEN       STRINGLAENGE GEWINNEN
       SRL  LEN,8             BYTE-->WORT
       JEQ  DSPRT             LEERSTRING --DANN ZURUECK
*
       LI   R2,DASTRG+1       STRINGADRESSE
DSPYLP MOV  R3,R0
       A    @COL,R0           SCREENTABLE OFFSET
       MOVB *R2+,R1           ZU JEDEM ASCII-CODE WIRD 96 ADDIERT
       AB   @BYT96,R1         UM ZUM BASIC PATTERN DESCRIPTOR TABLE
       BLWP @VSBW             OFFSET ZU GELANGEN
*
       DEC  SIZE              ALLE POSITIONEN GEDRUCKT?
       JEQ  DSPRT             JA
       DEC  LEN               STRINGENDE?
       JEQ  DSPRT             JA
       INC  @COL
       C    @COL,@H30         GROESSTE SPALTENPOSITION IST 30
       JLT  DSPYLP
*
DSPRT  B    @BSRTN            BASIC RETURN
*
ACCEPT LWPI MYWS              EIGENE ARBEITSREGISTER LADEN
       LI   R0,>1500          WERT FUER EINGABEVERZOEGERUNG
       MOV  R0,@DLYTME        NACH DLYTME UEBERTRAGEN
       BL   @GETRCS           ROW,COL,SIZE FUER ACCEPT UEBERNEHMEN
*
COACPT MOV  @COL,R6
       LI   R7,1
ACPTLP BL   @DELAY            PAUSE FUER EINGABEVERZOEGERUNG
       MOV  R3,R0             SCREEN TABLE OFFSET
       A    R6,R0             IN R0
       BL   @KINPUT           TASTATUR ABFRAGEN
*
       CB   @KEYBRD,@LEFT     TASTE"<--" GEDRUECKT?
       JNE  TRIGHT            NEIN
       C    R6,@COL           LINKES ENDE ERREICHT
       JEQ  ACPTLP            JA DANN HALTEN
       DEC  R6                NEIN ->CURSORPOSITION ERNIEDRIGEN
       DEC  R7
       JMP  ACPTLP
*
TRIGHT CB   @KEYBRD,@RIGHT    TASTE"-->" GEDRUECKT
       JNE  TERASE             NEIN
       C    R6,@H29            RECHTER ANSCHLAG ERREICHT
       JEQ  ACPTLP             JA DANN HALTEN
       C    R7,SIZE            BUCHSTABENZAHL ERREICHT?
       JEQ  ACPTLP             JA POSITION HALTEN
       INC  R6                 NEIN POSITION ERNIEDRIGEN
       INC  R7
       JMP  ACPTLP
*
TERASE CB   @KEYBRD,@ERASE    TASTE ERASE  GEDRUECKT
       JNE  TCLEAR             NEIN
       BL   @DSIZE             EINGABEPOSITIONEN WIEDER LOESCHEN
       JMP  COACPT
*
TCLEAR CB   @KEYBRD,@CLEAR    TASTE CLEAR  GEDRUECKT
       JNE  TENTER             NEIN
       B    @BADARG            JA DANN  BAD ARGUMENT ANZEIGEN UND STOP
*
TENTER CB   @KEYBRD,@ENTER    TASTE ENTER GEDRUECKT
       JEQ  GETSTR             JA STRING UEBERNEHMEN
*
       AB   @BYT96,@KEYBRD     SCREENCHARACTER FUER BASIC
       MOVB @KEYBRD,R1
       BLWP @VSBW              CHARACTER ANZEIGEN
*
       C    R6,@H29            RECHTER RAND ERREICHT
       JEQ  ACPTLP             JA
       C    R7,SIZE            CHARACTERANZAHL ERREICHT
       JEQ  ACPTLP             JA
       INC  R6                 NAECHSTE SPALTE
       INC  R7                 NAECHSTE ZEILE
       JMP  ACPTLP
*
*CHARACTER AUS SCREEN UEBERNEHMEN
*
GETSTR LI   R2,DASTRG          ADRESSE DES STRINGS
       MOVB @H00,*R2+          CHARACTERZAHL=0
GSTRLP MOV  R3,R0              SCREEN TABLE OFFSET BERECHNEN
       A    @COL,R0
       BLWP @VSBR              CHARACTER LESEN
       AB   @BYTM96,R1         NACH ASCII TRANSFORMIEREN
       MOVB R1,*R2+            IN STRING EINLESEN
       AB   @BYT1,@DASTRG      CHARACTERZAHL ERHOEHEN
       DEC  SIZE               NAECHSTES BYTE
       JEQ  ACPTRT
       INC  @COL              NAECHSTE SPALTE
       C    @COL,@H30          LETZTE MOEGLICHE SPALTE ERREICHT
       JLT  GSTRLP
*
ACPTRT MOVB @H00,@STATUS       STRING UEBERGEBEN
       CLR  R0
       LI   R1,4               STRING IST 4.PARAMETER IN LINK
       LI   R2,DASTRG
       BLWP @STRASG            STRING ZUWEISEN AN LINK PARAMETER
*
       BL   @BSRTN             ZURUECK  ZUM BASICPROGRAMM
******************
*TASTATUR ABFRAGEN*
******************
KINPUT MOV  R11,@RTLEV1        RETURNADRESSE SICHERN
       MOVB @H00,@>8374
       BLWP @VSBR              MOMENTANEN SCREEN CHARACTER LESEN
       MOVB R1,@SCRCHR
*
KEYLP1 LI   R15,300            ZUERST CURSOR ANZEIGEN
```

```
        MOVB @CRSR,R1
        BLWP @VSBW
KEYLP2  BLWP @KSCAN             TASTATUR ABFRAGEN
        CB   @KEYBRD,@HEXFF     KEINE TASTE GEDRUECKT
        JNE  KEYRT              WENN DOCH DANN ZURUECK
        DEC  R15                WIEDER ABFRAGEN
        JOC  KEYLP2
*
        LI   R15,350            ZWEITE SCHLEIFE
        MOVB @SCRCHR,R1         VOHER GELESENEN CHARACTER ANZEIGEN
        BLWP @VSBW
KEYLP3  BLWP @KSCAN             TASTATUR ABFRAGEN
        CB   @KEYBRD,@HEXFF     KEINE TASTE GEDRUECKT?
        JNE  KEYRT              WENN JA DANN ZURUECK
        DEC  R15
        JOC  KEYLP3
        JMP  KEYLP1             NOCH EINMAL DAS GANZE
*
KEYRT   MOVB @SCRCHR,R1         SCREENCHARACTER WIEDER ANZEIGEN
        BLWP @VSBW              DADURCH BLEIBT DER CURSOR NICHT
        MOV  @RTLEV1,R11        HAENGEN WENN ENTER GEDRUECKT WIRD
        RT
*
*
DSIZE   MOV  R11,@RTLEV3        RETURNADRESSE GEWINNEN
        MOV  @COL,R4
        MOV  SIZE,R5
        LI   R1,>8000           LEERZEICHEN ASCII 32
DSZELP  MOV  R3,R0              SCREEN TABLE OFFSET BERECHNEN
        A    R4,R0              UND BYTE FUER
        BLWP @VSBW              BYTE LOESCHEN
        DEC  R5
        JEQ  DSZERT
        INC  R4
        C    R4,@H30            LETZTE SPALTENPOSITION ERREICHT
        JLT  DSZELP             NEIN DANN WEITER
*
DSZERT  MOV  @RTLEV3,R11        RETURNADRESSE GEWINNEN UND
        RT                      ZURUECK
*
GETRCS  MOV  R11,@RTLEV2        RETURNADRESSE SICHERN
*                               ROW,COL,SIZE UEBERNEHMEN
SIZE    EQU  9
LEN     EQU  10
*
        LI   R1,1               1.PARAMETER IST ROW(ZEILE)
        MOV  R1,@PRMTR
        BL   @GETVAL
        DATA 1,24,ROW
*
        INC  @PRMTR             2.PARAMETER IST COL(SPALTE)
        BL   @GETVAL
        DATA 1,28,COL
        INC  @COL
*
        INC  @PRMTR             3.PARAMETER IST SIZE(GROESSE)
        BL   @GETVAL
        DATA -28,28,MYWS+18     ZIEL IST SIZE ALSO R9=MYWS+18
*
        DEC  @ROW               ZEILENPOSITION IM SCREEN IMAGE TABLE
        MOV  @ROW,R3
        SLA  R3,5               (ROW-1)*32
*
        MOV  SIZE,SIZE          SIZE NEGATIV?
        JEQ  RCSRT2             0 POSITIONEN -->BEENDEN
        JLT  RCSRT1             SIZE IST NEGATIV
        BL   @DSIZE             SIZE POSITIONEN LOESCHEN
RCSRT1  ABS  SIZE
        MOV  @RTLEV2,R11        RETURNADRESSE GEWINNEN UND ZURUECK
        RT
RCSRT2  BL   @BSRTN             BEI LEERSTRING ZURUECK INS BASIC
*
```

```
100 DIM TRICK$(100),HEX$(16)
110 CALL INIT
120 CALL LOAD("DSK1.BSCSUP",
"DSK1.CCAD/OBJKT")
130 CALL SCREEN(13)
140 CALL CLEAR
150 RESTORE 2300
160 GOSUB 650
170 RESTORE 2340
180 GOSUB 710
190 GOSUB 770
200 GOSUB 1120
210 GOSUB 1240
220 GOSUB 540
230 CALL LINK("DRAW",2,3,17,
18,Z1,S1,1000,60,Z11,S11)
240 GOSUB 620
250 GOSUB 560
260 CALL VCHAR(2,20,32,19)
270 CALL LINK("SELECT",2,20,
13,20,2,21,1500,Z2,S2)
280 CALL HCHAR(Z2,20,30)
290 ON Z2-1 GOTO 230,300,360
,360,360,360,360,360,360,360
,360,360,360
300 RESTORE 2380
310 FOR I=1 TO 2
320 READ SCZ1,SCS1,SCZ2,SCS2
,RTNG,SCOUNT
330 GOSUB 1060
340 NEXT I
350 STOP
360 ON Z2-3 GOSUB 420,390,86
0,880,930,930,930,930,2100,1
300
370 GOSUB 560
380 GOTO 260
390 GOSUB 440
400 GOSUB 570
410 RETURN
420 CALL LINK("INVERT")
430 RETURN
440 CALL LINK("CHRSCN",A$)
450 IF TCT>=100 THEN 460 ELS
E 520
460 FOR I=1 TO 4
470 CALL LINK("DSPLAY",21,1,
18,"BILDSPEICHER VOLL!")
480 GOSUB 1090
490 CALL HCHAR(21,3,32,18)
500 NEXT I
510 RETURN
520 TCT=TCT+1
530 TRICK$(TCT)=A$
540 CALL LINK("DSPLAY",22,21
,3,STR$(TCT))
550 RETURN
560 CALL LINK("CHRSCN",A$)
570 CALL CHAR(156,SEG$(A$,1,
16))
580 CALL CHAR(157,SEG$(A$,17
,32))
590 CALL CHAR(158,SEG$(A$,33
,48))
600 CALL CHAR(159,SEG$(A$,49
,64))
610 RETURN
620 Z1=Z11
630 S1=S11
640 RETURN
650 READ Z1,S1,A
660 FOR I=1 TO A
670 READ ZE,SP,VAR$
680 CALL LINK("DSPLAY",ZE,SP
,LEN(VAR$),VAR$)
690 NEXT I
700 RETURN
710 READ A
720 FOR I=1 TO A
730 READ NR,CR$
740 CALL CHAR(NR,CR$)
750 NEXT I
760 RETURN
770 CALL HCHAR(1,3,60,16)
780 CALL VCHAR(2,2,62,16)
790 CALL VCHAR(2,19,62,16)
800 CALL HCHAR(18,3,60,16)
810 CALL HCHAR(18,22,156)
820 CALL HCHAR(19,22,157)
830 CALL HCHAR(18,23,158)
840 CALL HCHAR(19,23,159)
850 RETURN
860 LC=126
870 GOTO 890
880 LC=32
890 FOR I=2 TO 17
900 CALL HCHAR(I,3,LC,16)
910 NEXT I
920 RETURN
930 RTNG=Z2-7
940 CALL LINK("DSPLAY",20,1,
28,"SCROLL SEKTOR WAEHLEN!")
950 SCOUNT=1
960 ON RTNG GOTO 970,970,102
0,1020
970 CALL LINK("SELECT",2,3,1
6,18,2,3,1000,SCZ1,SCS1)
980 GOSUB 1090
990 CALL LINK("SELECT",SCZ1+
1,SCS1,17,18,17,18,1000,SCZ2
,SCS2)
1000 GOSUB 1090
1010 GOTO 1060
1020 CALL LINK("SELECT",2,3,
17,17,2,3,1000,SCZ1,SCS1)
1030 GOSUB 1090
1040 CALL LINK("SELECT",SCZ1
,SCS1+1,17,18,17,18,1000,SCZ
2,SCS2)
1050 GOSUB 1090
1060 CALL LINK("SCROLL",SCZ1
,SCS1,SCZ2,SCS2,RTNG,SCOUNT)
1070 CALL HCHAR(20,1,32,32)
1080 RETURN
1090 CALL SOUND(200,444,0)
1100 CALL SOUND(1,30000,0)
1110 RETURN
1120 RESTORE 2390
1130 FOR I=1 TO 16
1140 READ HEX$(I)
1150 NEXT I
1160 READ CHAR$
1170 POS$="0123456789ABCDEF"
1180 FOR I=1 TO 32 STEP 2
1190 VAR$=HEX$(POS(POS$,SEG$
(CHAR$,I,1),1))&HEX$(POS(POS
$,SEG$(CHAR$,I+1,1),1))
1200 VAR$=VAR$&HEX$(POS(POS$
,SEG$(CHAR$,I+32,1),1))&HEX$
(POS(POS$,SEG$(CHAR$,I+33,1)
,1))
1210 CALL LINK("DSPLAY",INT(
I/2)+2,1,16,VAR$)
1220 NEXT I
1230 RETURN
1240 GOSUB 560
1250 FOR I=1 TO 4
1260 CALL LINK("SCROLL",2,3,
17,18,I,4*I)
1270 NEXT I
1280 GOSUB 560
1290 RETURN
1300 CALL LINK("SCRSAV")
1310 CALL POKEV(768,00,00,00
,00)
1320 IF TCT=0 THEN 1330 ELSE
 1380
1330 CALL CLEAR
1340 CALL LINK("DSPLAY",10,1
,28,"BILDSPEICHER IST LEER!!
!!!!")
1350 CALL SOUND(3000,196,0,2
62,0,330,0)
1360 CALL SOUND(1,30000,30)
1370 GOTO 1460
1380 RESTORE 2360
1390 READ A
1400 CALL CLEAR
1410 GOSUB 560
1420 GOSUB 660
1430 GOSUB 810
1440 CALL LINK("SELECT",4,6,
9,6,3,6,1500,Z2,S2)
1450 ON Z2-3 GOTO 1490,1720,
1460,1780,1970,2050
1460 CALL LINK("SCRRET")
1470 GOSUB 540
1480 RETURN
1490 IF TCT=0 THEN 1440
1500 RESTORE 2410
1510 READ A
1520 GOSUB 660
1530 TCT1=0
1540 FOR BILD=1 TO TCT
1550 A$=TRICK$(BILD)
1560 CALL LINK("DSPLAY",20,1
1,3,STR$(BILD))
1570 GOSUB 570
1580 CALL SOUND(10,1000,0)
1590 CALL LINK("SELECT",22,4
,24,4,22,4,1500,Z3,S3)
1600 ON Z3-21 GOTO 1630,1610
,1640
1610 TRICK$(BILD)=""
1620 TCT1=TCT1+1
1630 NEXT BILD
1640 FOR I=1 TO TCT
1650 IF TRICK$(I)="" THEN 16
60 ELSE 1680
1660 TRICK$(I)=TRICK$(I+1)
1670 TRICK$(I+1)=""
1680 NEXT I
1690 TCT=TCT-TCT1
1700 CALL HCHAR(20,1,32,165)
1710 GOTO 1440
1720 FOR BILD=1 TO TCT
1730 A$=TRICK$(BILD)
1740 IF A$="" THEN 1760
1750 GOSUB 570
1760 NEXT BILD
1770 GOTO 1440
1780 DELETE "DSK1.CHARCODE"
1790 OPEN #1:"DSK1.CHARCODE"
,DISPLAY ,VARIABLE 80
1800 FOR CN=1 TO TCT
1810 TEIL$=TRICK$(CN)
1820 FOR TEIL=1 TO 49 STEP 1
6
1830 VA$(INT(TEIL/16)+1)=SEG
$(TEIL$,TEIL,16)
1840 NEXT TEIL
1850 FOR D=1 TO 4
1860 DRUCK$=DRUCK$&"        D
ATA "
1870 FOR I=1 TO 16 STEP 4
1880 DRUCK$=DRUCK$&">"&SEG$(
VA$(D),I,4)&","
1890 NEXT I
1900 DRUCK$=SEG$(DRUCK$,1,LE
N(DRUCK$)-1)
1910 PRINT #1:DRUCK$
1920 DRUCK$=""
1930 NEXT D
1940 NEXT CN
1950 CLOSE #1
1960 GOTO 1440
```

BÖRSE

TI 99/4A
Wer hat noch Sportspiele? Bitte kurze Spielbeschreibung mit Liste der benötigten Geräte an
K. Wolters, Corneliastr. 9, 43 Essen 1, schicken!!!!

Dt. Handbuch für E/A zu verkaufen. W. Götz, 0711/864053

Suche: externe ansteckbare 32 k-RAM, Sprachsynthesizer und Buch Rogers Modul, K. Wolters' 0201/772141

Suche:
ext. 32 K-RAM Erweiterung, Buck Rogers Modul und Ex.-Basic (Mechatronic)
Zu melden bei K. Wolters, Tel. 0201/772141

Verkaufe: Schachmodul und TI-Extended Basic. Zu melden bei
K. Wolters, Telefon: 0201/772141

Suche dringend Kons. mit Rek. Kabel und Pal. Mod. bis DM 100. Tel. 3137650 (089) ab 19 Uhr)

Suche günst. Drucker mit V24-Anschl. incl. Kabel. Tel. 089/3137650 ab 19.00 Uhr

---

Günstig abzugeben!!! Ex-B. dt. Nachbau + 32 K + Cent. + Soft w. 500 DM. Tel. 02955/6546

Zu verk. orig. Ti-Ext. Basic Modul u. Lit., Spiele in Ext. Basic VB 190,– DM Tel. 06441/52556

Suche original TI-Joysticks bis 30 DM
Tel. (030) 7865425

Verkaufe Masch.Progr. (ED/Ass) auf Kassette) nur EX-Basic und 32 K. nötig. – Infos gegen Rückporto – B. Dusny, Mülldorfer Str. 31, 5205 St. Augsutin 1

Verkaufe:

Speech Synthesizer	120,–
Speech Editor	30,–
Terminal Emulator II	60,–
Mini Memoriy	160,–
Dateiverwaltung Modul	50,–
Invaders	25,–
Munchman	25,–
Parsec	30,–

Michael Strub, Hauptstraße 8774 Leuggelbach (CH)

****ACHTUNG****
Suche TI-User zwecks Spiele-Austausch (in Ex-Basci)
Bitte melden bei: Olav Keith, Tel. 0221/373530 *ab 19 h*

Tausche Prg. in TI+Ext. Basci! Liste an: H.-J. Eckers, Hermann-Hesse-Str. 7, 4050 Mönchengladb. 1, Interessenten für Clubbeitritt („TECHNOMANIA" aus MG) möglichst aus Raum D, MG, NE, KR; bitte ebenfalls bei mir schriftlich melden!!!

Wer verkauft günstig Schach-Modul, Sprachsynthesizer od. 32 k-Erweiterung? Angebote an: H.-J. Eckers, Hermann-Hesse-Str. 7, 4050 Mönchengladbach 1, Tel. 02161/52603

Verkaufe: Komplettes TI-99 System! Weitere Auskünfte: Oliver Siffrin, Tel. 06821/7522

```
1970 DELETE "DSK1.CHARMEM"
1980 OPEN #1:"DSK1.CHARMEM",
VARIABLE 163
1990 FOR CN=1 TO TCT
2000 PRINT #1:CHR$(0)&CHR$(C
N)&CHR$(147)&CHR$(200)&CHR$(
64)&TRICK$(CN)&CHR$(0)
2010 NEXT CN
2020 PRINT #1:CHR$(255)&CHR$
(255)
2030 CLOSE #1
2040 GOTO 1440
2050 FOR I=1 TO TCT
2060 TRICK$(I)=""
2070 NEXT I
2080 TCT=0
2090 GOTO 1440
2100 GOSUB 880
2110 CALL LINK("DSPLAY",2,1,
16,"BITTE CHAR/CODE")
2120 CALL LINK("DSPLAY",3,1,
16,"EINGEBEN!0-9/A-F")
2130 CHAR$=""
2140 FOR I=1 TO 4
2150 CALL LINK("ACCEPT",I+4,
1,-16,VAR$)
2160 FOR VCT=1 TO 16
2170 IF POS(POS$,SEG$(VAR$,V
CT,1),1)=0 THEN 2200
2180 NEXT VCT
2190 GOTO 2250
2200 CALL LINK("DSPLAY",16,1
,16,"EINGABEFEHLER!!!")
2210 CALL SOUND(500,444,0)
2220 CALL SOUND(1,30000,30)
2230 CALL HCHAR(16,3,32,16)
2240 GOTO 2150
2250 CHAR$=CHAR$&VAR$
2260 NEXT I
2270 GOSUB 880
2280 GOSUB 1180
2290 RETURN
2300 DATA 2,3,15,24,1,COMPUT
ER AIDED CHAR DESIGN,2,19,ZE
ICHNEN,3,19,STOP,4,19,INVERT
,5,19,CHARSAVE
2310 DATA 6,19,FUELLEN,7,19,
LOESCHEN
2320 DATA 8,19,SCRLDOWN,9,19
,SCRLUP,10,19,SCRLRIGHT,11,1
9,SCRLLEFT,12,19,C-EINGABE
2330 DATA 13,19,TRICKFILM,14
,19,&SPEICHERN,22,1,GESPEICH
ERTE BILDER=0
2340 DATA 7,156,000000000000
0000,157,000000000000000000,15
8,0000000000000000,159,00000
0000000000
2350 DATA 126,7E8199A5A59981
7E,60,00003C24243C,62,00003C
24243C0000
2360 DATA 7,1,1,--BILDSPEICH
ER BEARBEITEN---,4,5,BILDAUS
WAHL,5,5,FILM AB,6,5,ZURUECK
2370 DATA 7,5,SPEICHERN MERG
E FORMAT,8,5,SPEICHERN E/A F
ORMAT,9,5,SPEICHER LOESCHEN
2380 DATA 1,1,24,16,1,24,1,1
6,24,32,2,24
2390 DATA "     ","   ~","  ~
 ","  ~~","  ~  ","  ~ ~~","  ~~
 ","  ~~~","  ~   ","  ~~  ~~","~~ ~
 ","~~ ~~","~~   "

2400 DATA "~~ ~~","~~~~ ",~~~~
,00F04848484848F0000F08080E0
8080F88D8A888888888000000F090
909090900F
2410 DATA 4,20,3,BILD NR.,22
,3,BILD OK,23,3,BILD LOESCHE
N,24,3,ZURUECK
```

BÖRSE

Graphik-Tabl. 150 DM, Flugsim (Cas) 35 DM, TI-Modul American Football Sup Graphik 65 DM Kopf 095116846

TI-Tunnel's of Doon dt. Übersetzung für 15,– DM Vorkasse. M. Redlich, Eichenweg 3, 4620 Castrop-Rauxel 2

Suche preisgünstig TI-Writer, Multiplan
Dieter Redlich, Eichenweg 3, 4620 Castrop-Rauxel 2

Suche Anwenderprogramme. Verkaufe etliche Module + Erweiterungen. Horst Nietowski, 02173/15395 Langenfeld

---

Verkaufe: original TI-EX-Basic. Tel. 040/3193417 Anruf nach 18 h

---

Ext.-Basic + XB-Lehrg. + 50 XB-Prgrm. 250 DM, 10 Module (Schach, M.-Maker usw.) ab 15 DM, TI-Bücher ab 9 DM, Telefon 02174/40654

TI-99A, RS232 ext. 32k, P-Box incl. Disk, etl. Module, einzeln zu v.
K. Berdon, Odenwaldstr. 12, 6056 Heusenstamm
Tel. 0610462185

Verkaufe Ext.-Basic für VB 160,– DM + Datenverwaltung & Analyse VB 45,– DM + Othello VB 20,– + Parsec VB 40,– DM + Konsole (etwas defekt) VB 60,– DM!
Matthias Orf, Birkenallee 34, 3507 Baunatal 1,
Tel. 0561/497990

TI-CLUB BAUNATAL bietet: Clubheft mit 25 Seiten, Prg.-Speicher (400 Programme!), Entfernen von Listschützen, Drucker, toller Ausweis für 2,– DM pro Monat!! Info gegen 50 Pf. oder aktuelles Clubheft gegen 3,– DM anfordern bei: TCB, Matthias Orf, Birkenallee 34, D-3507 Baunatal 1, Tel. (0561) 497990

TI-CLUB BAUNATAL sucht Kontakt zu anderen Ti-User-Clubs zwecks Info- und Gedankenaustausch.
TI-CLUB BAUNATAL, Matthias Orf, Birkenallee 34, D-3507 Baunatal 1, Tel. 0561/497990

TI-32K Speichererweiterung f. Modul Box zu verkaufen 300 DM 06441/74830

HALLO 99' User! Wie, Sie sind noch nicht im MONSTERVISION Club! Dort gibt es monatlich ein 32seitiges Magazin mit vielen Vorteilen für alle Mitglieder. Eine Clubgebühr ist nicht vorhanden! Gratis-Infos bei: MV Club, Gratis-Info. Uesener Ring 30, 2807 Achim

---

TI99/4A + Box + Disk + Ext Basic + Sp. Synth + Chess + Invaders + Joysticks + Literatur DM 1100.
Tel. (SA, SO) 09621/85143

Verk. TI+Rec Kabel + Rekorder + Joaystick + 5 Module + Basic Lehrgang + Literatur für 340 DM.
Tel. 06721/43307

Verkaufe: Ti + Ext. Basic + Parsec + Music-Maker + viel Software für nur 250,– DM Ideal für Einsteiger
Markus Schröpfer, Schulstr. 22, 8567 Neunkirchen

Suche TI-User Club in Berlin (West) zwecks Informations-Austausch.
Ab 19 h. Tel. 811 51 46

---

6 Top-Hits direkt vom Autor. Z.B. Nanuk der Eskimo, Stardust, Alien-Landing. Senden Sie 30,– DM im Kuvert an: W. Döltsch
A.D. Hinterstein 10
6108 Weiterstadt 3

Letzte Lösung! Brauche dringend Geld. Verk. TI99/4A + TI XBasic + Bas. / XBas. Lehrpr. + Lehrbücher / Lit. + Joy Adapter + 2fach Rec. Kabel + Chis. Trail + ca. 280 Progr. Alles gepfl. VB 450 DM. Tel. 0231/875916 ab 18 h

---

Wegen Syst.auflösung zu verk. Org. TI RS232/V 24 100,–, Speechs. 70,–, Module von 20,– bis 50,–, Compiler 100,–, Atari Module Jungle, Dig, Kong je 40,– Tel. 09151-95153

Suche Heft Computer Praxis 12/84 in einigermaßen gutem Zustand. Tausche gegen 100 EXB (Tib) Prg's oder zahle bis zu 5 DM. Verkaufe engl. Handbuch für das EXBModul für 15 DM.

Tausche Programme in TI und Ex-Basic. Habt Ihr Lust? Meldet Euch einfach bei mir: 07156/34 941 od. schriftl. Carlos Jarque, Ludwigsburger Str. 14 / 7257 Ditzingen 1

ZU VERSCHENKEN habe ich nichts ... aber da ich mir eine 32K Erweiterung kaufen möchte, bin ich gezwungen, etwas aus meiner Computersammlung zu verkaufen. Z.B. 11 Adventure Cass. original TI Stück 15,– DM., M*A*S*H* 40,– DM, TI Bücher 20–30 DM, Computer hefte z.B. HC, Computronic u.s.w. 2,– 50 – 3,– DM, Spielmodule für ATARI Compuer z.B. GYRUSS, STAR-TRECK, Popeye, O-BERT u.s.w. 30–50 i DM. Info gegen frankierten Rückumschlag bei: Manfred Lipowski, In der Wanne 165, 462 Castrop-Rauxel 4, Tel. 02305/72237

Komplett TI99/4A Konsole + PBox + RS232 + Ext. Basic + Laufwerk + 32 Kram + Minimemory + Schach + Lit. + Softw.
Tel. 07231/41436

Hey TI-Freaks! Wer hat Lust mit Nappsoft Programme in TI + Ex. Basic zu tauschen? Liste an Martin Roth, Bruckner Str. 3, 6680 Neunkirchen 7

Verkaufe TI99/4A+dt.X-Basic+orig. TI-Box m. Controller+Laufwerk+Spielmodule+Joystickadapter+ viel Software auf Disk. wegen Systemwechsel. Abgabe gegen Höchstgebot!
C. Reusch, Tel. 0241/172129

Hallo TI E/A User. Wer von Euch hat Lust, mit mir Prg's zu tauschen. Markus Junghans. Tel. 06106/74182

Verkaufe TI99/4A + EX + 14 Module + Literatur + Zubehör. Alles original verpackt. Tel. 02174/40616 nach 15 Uhr

---

Schüler sucht für TI99/4A 32 K Ext und Ext. Basic o. Exb. II + M. Stief/Sandstücke 21/28 Bremen 61

Achtung TI-Adv. Freunde!!! Jetzt gibt's DAS Graphic-Adv., auf das Ihr schon lange gewartet habt. (ca. 16k) TI-Basic; kein Adv.-modul notwendig. Für DM 5 + Vers. Bei J. Laux, Schulgartenstr. 20, 6638 Dillingen. Es lohnt sich

---

Verk. Ex. Basic + Handbuch + 100 Super Programmen für nur 200 DM. Ralf Ludwig, Am Mergelsberg 31, 4000 Düsseldorf 12, Tel. 0211/297042

994A+Ex.B. (engl. u. Deu.H. buch) + Kass. rec. + Kabel + Spech-Syn. + Munchman+ Engl. Grammatik+Literatur 600,–. Tel. 069/554026
T. Veith

Tausch*Musik Maker* Datenverw. u. Analyse* Comp. Kurs Bis Nr. 56* U.A.M. * Su. Extern Erw. * Assembler * U.A.Ang.*

Schnittstelle für Direktanschluß eines Druckers sowie viele tolle Module-Preis VB Tel. 06103/72518

Verkaufe meine Spielesammlung (ca. 300–350 Prg's; 70 % EXB, 30 % Tib) an den am meist bietenden!!! Übernehme Porto + Cassette N (15 x C 60)! Tel. 07156/34941 oder Carlos Jarque, Ludwigsburgerstr. 14, 7257 Ditzingen 1

Verkaufe 99/4A mit Box, Dsk, 32K, X-Basic, E/A, dt. E/A Kurs, Statistik, Modulexpander und viel Literatur. Nur komplett! Neu über 3000,– jetzt 1600,– 05300/485 Uli

---

Verkaufe ExBasic + Adventure Modul + 12 Cassetten +Literatur für 250 DM Tel. 08638/67495 ab 19 h

---

Verk. Assemblerkurs: Asem-4 Band 1 + 2 + Disk DM 50,– C. Kater 7, Rue de Schoenfels, L-7432 Gosseldange Tel. 328060

Suche HARDCOPY-Programm für EPSON-Drucker. Alexander Rupp, Kalmanstr. 45, 6600 Saarbrücken; Tel.: 0681/45134

Suche Schaltungsunterlagen für TI-Druckerinterface sowie Schnittstelle RS232. R. Schinkel, Hamburger Str. 20717, 2200 Elmshorn

---

Verk. TI99/4a-Konsole + Netzteil + 32 K extern incl. durchgef. Bus u. Centronic + Ext-Basic Modul + Rec. Kabel + Centronic-Kabel + Joyst.-Adapter + Literatur einzeln od. zusammen, Preis: VHS
Tel. 05283/1850 Mo–Mi 18 h

Verk. TI und Ex Basic Programme. Info gegen Rückporto. B. Knedel, Tulpengasse 16, 3171 Weyhausen, T. 05362/71187

BÖRSE

Verkaufe kompl. für 400 DM TI99/4A anschlußfertig + 1 Modul Ex.-Ba. mit Handbuch + Cas.-Rec. M. Kabel, 3 Bücher, 6xTI-Journal + 8xTI-Revue + 1 Monitorkabel. Kl. Entinger, 6650 Homburg, Westring 22, Tel. 06841/71693 nach 17 h

---

Verkaufe 20 TI-Progr. Stck./ 8,– DM wie z.B. Flugsimulator, Star-Strike, Uambler etc. Ruft an: 06874/6705

TI99/4A Speechsynt. 2 Joyst. 12 Module (ext. Bas., Parsec, Schach ...) Kassetten, viel Literatur 500,– DM Tel. 06834/41660 (Klose)

Verkaufe TI-Writer + Multiplan zus. DM 250,–. Tel. 04262/1205

TI99/4A; neuwertig; Handbuch, orig. Verp. + Recorderkabel: 170 DM. B. Dobrick, Hohenstaufenstr. 17, 7340 Geislingen

---

Wer im Raum Hi/H/BS hat CPS 99 Atronic. Bitte melden. Tel.: 05069/6538. Rufe zurück. JORK Warnecke

---

Suche Bedienungsanleitung zur Diskvers. des X-Basic II. Wer kann helfen? Tel. 02051 66950 ab 20 Uhr.

TJ Expannsion Box ller mit Netzteil DM 120 Diskgehäuse mit Kabel + Netzt. DM 50 zu verk. 08233/6653

Verk. (orig. TI) Diagnostic-Mod + RGB /Modulator (H4/85) Stulen Roerdomps 7671WL Vriezenveen, Tel. 0933549962650

Suche PBox + Disklw (ev. nicht Org. TI) + 32K + RS 232 + evTI Writer + Speech-Synth + mit Handbüchern Tel. 0711 843775

Verk. orig. Ex-Basic, unbenutzt. suche Speech-Synth. + TE 2, Frank Brengel, 089/1231332 ab 18 Uhr

---

Suche PAL Modulator PHA 2036 for TI-99A Channel 36 auch Spielprogramme.

Suche TIBox mit Contr/Ext. DSKContr/Sprachsynthesizer/Avend u. ExModul. Dieter Wagner, 7750 Konstanz, Leipzigerstr. 9

Verkaufe Module: Speech Editor, Attack, Ext. Basic, etc. Tel. 04131-55457 nach 19 Uhr

---

Tippe Listings ab. 10 DM (inc. Postg.) in Umschlag mit Listing abschicken. Schicke alles + Kass. zurück an: Aronica Luigi, Deeler Weg 14, 5000 Köln 71

Verkaufe wegen Systemerweiterung 3 original TI-Laufwerke, einen Diskcontroller und eine 32-KByte-Speichererweiterung. Liebald Heiko, Eichenweg 7, 7914 Pfaffenhofen o.d. Roth

---

Viel Zubehör für TI abzugeben wegen Systemaufgabe. Z.B.: Printer/Plotter, externe Cent. Schnittstelle (mit Kabel und durchge. Bus), original Joystick Ext. Basic, alle Adventure; Mini Assembler, Schachmeister; Moon Patrol und vieles mehr. Informationen bei Thorsten Rauer, Beetstr. 64, 4902 Bad Salzuflen, 05222/ 13182

Assembler-Programme f. E/A, XB, MM + 32 K: GPL-Disassembler; FAST-COPY (Sektorkopierer) kopiert jede Disk in drei Durchgängen. Info gg. Rückumschlag. Alles sofort lieferbar! M. Eichhorn, Ziegelheck 1, 6240 Königstein 4

Verkaufe: TI99/4A Konsole + Extended-Basic II plus 2 Module (Microsurgeon und Schachmeister) + Recorderkabel + Joystickadapter + 2 Joysticks + Prgm's + Hefte VB: DM 500,– ÖS 3500,– T.: Österreich / 0662/26671 (Salzburg)

Suche ext. Laufwerk + Disc-Kontroller günstig. Tel. 06805/8393

Verk. Minimem m. dt. Handbuch 190,– Pers. Rep. gen. Othello, Parsec, Stat., Ger., Householdbudg. Man. je 20,– J. Kupzig 02208/4165

STOP! Habt ihr ein TI zuhause stehen? Habt Ihr Lust, mit mir PGM's zu tauschen? Oder wollte Ihr sie kaufen? Oder sucht Ihr andere TI-User? Dann schreibt mir! M. Kugelmann, Nassauerstr. 4, 6272 Niederhausen!

Einmalige Chance! Junger TI-Freak möchte mit dir PGM's tauschen! Mords Software in TI, Ex-Basic oder Exbasic + RAM-Pack! Stelle auf Wunsch auch PGM's her! Adr.: M. Kugelmann, Nassauerstr. 4, 6272 Niederhausen!

Verkaufe TI99/4A + P-Box + 32K + Controller + Disklw. + Ex. Centrois + Joystik + ElA + d.t. Handb. + Ex-Basic + dt. Hb. + 6 Sp. Module + dv. TI Spezialb. + Super Grafic + d.v. Programme VB 1500,–. Jan Böhme, Röweland 16, 2000 HH 62

Verk. TI99/4A, P-Box, 32 K Erw., RS 232, Disk-Contr., Disk-Laufwerk, Sprachsyn., Ex.-Basic, Editor/Assembler, Mini-Memory, div. Module, TOP-Software (M.-Code), Literatur usw. VB 2000,– DM. Tel. 07132/37608

Verk. TI-99/4A + Joystick + Datasette + ca. 120 Programme + 6 Module + Fachliteratur. VB 280,– DM. M. Schreiner. Tel. 06103/ 81815

Verk. Disk-Contr., Laufwerk–orig. TI für P-Box. Suche Peri-Box orig. TI. Angebote an: Richert, 02921/2626

Verkaufe TI 99/4A + X-Basic + P-Box m. Laufwerk + Schnittstellenkarte + Datenverw. + Statistik + Recorder m. Kabel + Literatur für DM 1500 VB. Telefon: 06403/71104

NEU: Editor / Assembler auf Kassette: Nur 70 DM! Super schnell und komfortabel! Tel. 0561/887129: Lothar Krauß

Verkaufe TI/Ex-Basic für Anfänger und Fortgeschrittene. Tel. 0211/422216

Verkaufe original TI-Ex-Basic Handbuch (in englisch) Tel. 0211/422216

Achtung-TI-User. Verkaufe org. Ex Basic VB 165 DM. Angebote an Lukas Merten, Marktstr. 19, 5440 Mayen

---

Suche Original E/A Handbuch. Reimund Müller, Buchenstr. 20, 4630 Bochum 6, Tel. 02327/83764

Suche Modulexpander u. ausführliche Bedienungsanleit. für ext. 32K-Speichererwe. mit Centronic-Schnittstelle (auch Kopie) Angebote an: Berth. Fella Hardenbergstr. 28, 8500 Nbg 20, Tel. 0911/549804 (rufe zurück!)

Verk. für TI/99 4A BASF Laufwerk 6106 VB 300 DM Tel. 0208/73042

Verk. TI99/4A + Ext. Basic Modul, Preis VB. Tel.: 02151/405869

Verk. TI99/4A incl. Handbuch für DM 120,–. Alexander Rupp, Kalmanstr. 45, 6600 Saarbrücken. Tel.: 0681/45134

Verkaufe TI-99 100 DM
Exd. Basic + d. Handbuch 150 DM
Parsec (Mod.) 30 DM
TI-Schach (Mod.) 50 DM
Blasto (Mod.) 8 DM
Orig. TI-Recorder + Kabel 60 DM
Joysticks 10 DM
Thomas Schmidt, Eiswiese 7, 5160 Düren
Tel. 02421/42488

TI99/4A (def) + Orig. XBasic + deut. Handbuch + Rec. Kabel + Literatur. Preiswert! G. Schulte, 0421/71149 ab 17 h

---

Verk. TI-Konsole + XBasic + IMB-Interface + 64K-Dram-Karte + Zenith-Monitor VB 950 TI59+PC100C VB300 Tel. 0941/67884

Suche Peribox f. TI99/ 4A Diskettenlaufwerk 32K Erw. Artur Kasimir, Weslarnerweg Nr. 7, 4770 Soest, Tel. 02921/8591 nach 18 h

---

Drucke Listings kosten nur Rückporto beilegen. Verkaufe Spkel 1+2, Basic Inf. Marketing Plan Spiel 10,– DM

Suche ext. Laufwerk + Disc-Kontroller günstig. Telefon 06805/8393

TI 99/4A * Verk. Super Sketch (Grafiktablett) Markus Schenk. Tel. 07136/ 4116 ab 17 h

---

!!!Extended Basic!!! Verkaufe TI99/4 + Extended Basic + Handbuch + Recorder-Kabel (340,– DM) 0711/ 537789

Extended Basic Modul 180,–
Disc Manager II Modul 110,–
Parsec 30,–
Siegfried Fränkl, 0941/96296 ab 16.30 tägl.

Wer hilft 99/4A-Fan i.d. DDR? Gesucht: billiges 32K RAM, Konsole z. ausschlachten, Schaltplan und Bauanltg. f. Hardware, def. Hardware jed. Art. Kontaktadresse: Otto 42 OB 1, Feldmannstr. 54, Tel. (ab 19 h) 0208/860892

20/64/128

Das unabhängige Commodore-Magazin

Commodore kontra Atari: Wer bietet mehr?

Im Test: Eprombrenner

Neu: Die Peek- und Poke-Kartei für VC 20 & C 64

Rund 30 Seiten Listings

NEU!

Einblick: Wie das Betriebssystem arbeitet

JETZT AN IHREM KIOSK

Ausblick: Was es in den USA alles gibt

Tips, Tricks & Kaufberatung

ANDREAS EHLERDING TECHNOLOGIE
Ein neuer Name, wenn es um Computer geht.

Er steht für Forschung und Innovation an der Nahtstelle von Psychologie, Medizin und Informatik. Trotz finanzieller Engpässe konnten wir bei unseren Computern von vornherein nicht auf kompromißlose Zuverlässigkeit und höchste Flexibilität verzichten. So fiel die Wahl auf den TI 99/4A. Sehen konnte der TI bisher noch nicht, doch dank unserem VIDEO DIGITIZER AET - VD 99 ist dieses Handicap nun beseitigt.

DER BLICK IN DEN SPIEGEL

TI 99/4A mit angeschlossenem Video Digizer AET;VD 99.
Vergrößerter Bildausschnitt, 64 x 48 Punkte,
16 Farb bzw. Stufen mit spezieller Software in druckbare
2-Farben Darstellung umgewandelt.

WANN WIRD IHN IHR TI TUN?

Technische Daten:
64000 Bildpunkte 64 Grauwertstufen · belegt keinen Speicherplatz im TI · 64 K eigenes RAM, auch als Erweiterung nutzbar 32 K CMOS-RAM (auf Wunsch zusätzlich integriert) · incl. Bildverarbeitungs-Software · Darstellung der Bilder auf dem Bildschirm 256 x 192 Punkte · Objekterkennung (lernen durch zeigen) · Grauwerthistogram · Kantendetektion · Ausschnittvergrößerung eines beliebigen 64 x 48 Punktebereiches in Hardcopyroutine für Epson FX 80/RX 80

1498,– DM incl. MWSt.

ANDREAS EHLERDING TECHNOLOGIE
BERATUNG ENTWICKLUNG FERTIGUNG SERVICE · NIEDERSACHSENRING 26 · D-3051 WÖLPINGHAUSEN 05037/744

Achtung!!!

RADIX

RADIX Bürotechnik
Rappstraße 13 · 2000 Hamburg 13
Tel. 040/441695 · Telex 213682 radix d
tägl. 10.00-12.30 + 13.30-18.30 Uhr
Sa. 10.00-13.00 Uhr
Verkaufsstelle Kiel: Ziegelteich 23 · 2300 Kiel 1

IHR TI-SPEZIALIST
hält für Sie bereit:
32 KB Erweiterung
extern Batterie gepuffert **258,–**

GPL-DISASSEMBLER
auf Diskette
notwendig 32 K und
Editor Assembler
zum Auflisten von 6 ROM-Module
59,–

Preisliste bitte anfordern

IMMER NEU UND AKTUELL FÜR TI 99/4A

EXTENDED-BASIC (Mechatronic)
mit deutschem Handbuch 199.90
EXTENDED-BASIC II PLUS mit deutschem Handbuch 299.—
= Extended-Basic + Grafik Extended-Basic (Apesoft) in 1 Modul

Umtauschaktion
Bei Bestellung eines EXTENDED-BASIC II PLUS vergüten wir Ihnen DM 70.— bei kostenfreier Zusendung eines original amerikanischen Extended-Basic-Moduls (elektrisch/mechanisch einwandfreier Zustand!!)
Sie zahlen nur noch 229.—

Umbauaktion (gilt nur für deutschen Lizenznachbau „Mechatronic"). Wir machen aus Ihrem EXTENDED-BASIC ein EXTENDED-BASIC II PLUS mit deutschem Handbuch für nur 98.—

32-k-RAM-ERWEITERUNG mit Centronic-Interface, Kunststoffgehäuse 190 x 110 x 60 mm zum seitlichen Anstecken an den Bus, der Bus wird nach rechts durchgeschleift, mit 5-V-Steckernetzteil 289.50*

Unser Paketpreis-Angebot
EXTENDED-BASIC II PLUS + 32-k-RAM-ERWEITERUNG, ohne Centronic-Interface für nur 499.50*

NEU! 128-k-RAM-ERWEITERUNG, mit Centronic-Interface und 5-V-Steckernetzteil 595.—

NEU **Die Weltneuheit: 128 kB — GRAM Preis ca. 750,—**
Lieferbar etwa Janur 1986

Preise in DM/Stück inkl. MwSt. · Technische Änderungen vorbehalten
Versand gegen Nachnahme oder Vorauskasse.

4-FARBEN-PRINTER-PLOTTER PP-A 4, Centronic-Schnittstelle, DIN-A 4-Format, Direktanschluß an 32-k- oder 128-k-RAM-Erweiterung 699.—

ANSCHLUSSKABEL
von 32-k- oder 128-k-RAM an PP-A 4 68.—

SLIM-LINE-LAUFWERK 5,25", 500-k-Byte-DS/DD
(z. B. TEAC FT 55 B) 399,90—

EINBAUSATZ für 2 Laufwerke in original TI-P-Box 95.—

DISC-STEUERKARTE (CorComp), DS/DD, für max. 4 Laufwerke 635.—

NEU! QUICK-DISC-FLOPPY (im Gehäuse), zum Direktanschluß an die Konsole, **keine Steuerkarte erforderlich,** 128-k-Byte-DS, für 2,8"-Disketten, mit 5-V-Steckernetzteil, identisch mit der bekannten MSX-Version 598.—

SEHR NEU DISKETTEN 2,8", 10er-Pack 95.—
TI-MAUS — die schnelle und komfortable Cursorsteuerung mit Software auf 5,25"-Diskette, mit 5-V-Steckernetzteil 296.—

NEU VIERSPANNUNGS-SCHALTNETZTEIL, +5 V, 4 A/±12 V, 0,3 A/–24 V, 0,3 A, primär getaktet, 35 Watt, MOS-Fet-Technik, extrem klein (80x125x32 mm), offene Bauweise, ideal zum Betrieb von Druckern, Monitoren etc. 345.—

*** Preissenkung — dank großer Nachfrage!**

albs-Alltronic G. Schmidt · Postfach 1130 · 7136 Ötisheim
Tel. 0 70 41 / 27 47 · Telex 7 263 738 albs

l + special + special + special

TI 99/4A SPECIAL

SONDERHEFT NR. 4/86

DM 14,80 / ÖS 124 / SFR 14;80

Superhirn
Tankcommander
TI-Pitfall
Verflixte Börse
War Games
Zahlenjagd
Zauberschloß
Burgenschlacht
Jungler
Kreuzworträtsel
Let's jump
Long live the King
Solo-Würfeln
Spielautomat
Adventure III
Alle Neune u.v.m.

Rund 150 Seiten Listings für den 99/4A

Das Magazin für TI 99-4A

special + special + special + s

HeGa-Soft
Software, Hardware und Literatur
Postfach 1201 8044 Lohhof

Deutsches Handbuch zum Editor/ Assembler		DM 98,–
TI-LOGO Kurzanleitung (28 S.)		DM 19,80
TI-LOGO ausführliches Buch (218 S.)		DM 59,–
TI-Forth Diskette + Handbuch		DM 148,–
TI-Forth Extended Basic Loader Disc		DM 25,–
Diskettenkatalog mit Autostart		DM 15,–
Preissenkungen für Software (alter / neuer Preis)		
iD Data	(119,–)	DM 75,–
iD Konto	(119,–)	DM 75,–
iD Profi Set (Konto+Data)	(198,–)	DM 140,–
Graphic Master	(99,–)	DM 65,–
3 D World	(99,–)	DM 65,–
NEUE Software:		
Macro Word Plus Textverarbeitung Benötigt Extended Basic, Speicher-erweiterung, Diskettenlaufwerk und Drucker		DM 145,–
Macroman (Spiel)		DM 40,–
Disassembler		DM 75,–

Alle früheren Angebote sind hiermit ungültig.
Lieferung ausschließlich per Nachnahme.
Versandkosten Inland: bis DM 100,–
Bestellwert: 5,–; darüber: frei
Versandkosten Ausland: jeweilige tatsächliche Postgebühren
Hinweis: iD Programme: Bestellungen bis 1.3.86 Auslieferung kurzfristig. Spätere Bestellungen Auslieferung ab Juli 86 w/Betriebsurlaub des Herstellers.

TI99/4A

PERIPHERIE

RS 232 Karte (orig. TI)	379,–
RS 232 Karte (Atronic)	299,–
32 K-Karte (Atronic)	299,–
Disccontroller DSDD (Atronic)	449,–
Disccontroller DSDD (Corcomp)	629,–
P-Code-Karte (orig. TI)	749,–
Compact Peripherie System CPS 99 mit 1 Diskettenlaufwerk DSDD + 10 Disketten	1399,–
CPS 99 mit 2 Diskettenlaufwerken DSDD + 10 Disketten	1749,–
Diskettenlaufwerk intern DSDD (Epson) mit Einbausatz	399,–
Externe 256 K-Erweiterung	589,–
Externe 32 K-Erweiterung	199,–
Ext. 32 K-Erw., batteriegepuffert	239,–
Externe 32 K-Erweiterung + 1 Centronicsschnittstelle	269,–
Externe 32 K-Erweiterung + Centronicsschnittstelle + Kabel + Epsondrukker LX 80	1239,–
dto. + Epsondrucker FX 85	1739,–
dto. + Stardrucker SG 10	1259,–
dto. + Stardr. Gemini 10 X	859,–
Sprachsynthesizer	189,–
Modulexpander, 3fach	125,–
RGB-Modulator	179,–
Akustikkoppler Dataphon S 21 d + externe V-24-Schnittstelle + Verbindungskabel	539,–
Ak. Dataphon S 21 d + Kabel für RS 232 Karte	299,–
TI-Maus, anschlußfertig	295,–
Joystickinterface + 2 Joysticks Quickshot II	89,–
Cassettenrecorderkabel	29,–
MBX-Sprachsteuereinheit + Baseballmodul, anschlußfertig	349,–

BÜCHER

Editor/Assembler Handbuch dt.	98,–
TI-Basic & Extended Basic dt.	48,–
Mini Memory Spezial dt.	55,–
TMS 9900 Assemblerhandbuch für das Mini Memory dt.	78,–
TI-99/4 A intern dt.	38,–

MODULSOFTWARE

Extendend Basic II Plus	289,–
Mini Memory (Orig. TI)	169,–
Editor/Assembler (32 K notw.)	179,–
TI-Writer (32 K notw.)	299,–
Multiplan (32 K notw.)	259,–
TI-Logo II (32 K notw.)	299,–
Diskfixer (Navarone)	149,–
Terminal Emulator II	85,–
Connect four, Attack, Yahtzee je	29,–
Alpiner, Car Wars, Chisholm Trail, Othello, Invaders, Munch Man je	39,–
Black Jack, Fathom, Hopper, Dig Dug, Defender, Soccer, Parsec je	49,–
Congo Bongo, Burgertime, Espial, Moonsweeper, Treasure Island, Bigfoot, Microsurgeon, Statistik je	59,–
Star Trek, Tunnels of doom, Music Maker, Jungle Hunt, Moon Patrol, Donkey Kong, Protector II je	69,–
Buck Rogers, Return to Pirate's Isle, Adventuremodul, Video Chess je	75,–
Datenverwaltung + Analyse	79,–
Popeye, Pole Position, Shamus je	89,–
Video Chess + Defender + Dig Dug + Attack + Fathom nur	175,–
Statistik + Donkey Kong nur	89,–
Defender + Munchman + Soccer	99,–

DISKETTEN- UND CASSETTENSOFTWARE

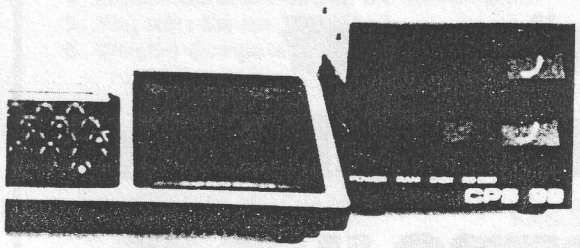
Preisliste mit Gesamtübersicht erhalten Sie gegen Zusendung eines Freiumschlags (Kennwort: TI-99/4 A).

Alle Preise inkl. MWSt. zuz. Versandkostenpauschale (Warenwert bis DM 1000,–/darüber): Vorauskasse (DM 8,–/20,–), Nachnahme (DM 11,20/23.20), Ausland (DM 18,–/30,–). Versand nur gegen Vorauskasse oder per NN; Ausland nur Vorauskasse. Gesamtpreisliste gegen Freiumschlag.

CSV RIEGERT
Schloßhofstr. 5, 7324 Rechberghausen, Tel. (07161) 5 28 89

Die nächste TI-REVUE erscheint am 25.April 86

TI-User: Wir helfen Ihnen



Peripherie

CPS 99 mit 1 Laufwerk DS DD (360 KB), Disk-Steuerkarte 32 K-RAM, RS-232 und Centronics	**1398,–**
256 K-Byte RAM (ext.)	**598,–**
32 K-Byte RAM (ext.) mit Centronics-Interface	**268,–**
Graphic Tableau	**148,–**

Software

Mini Memory	**198,–**
Editor Assembler	**165,–**
Assembler-Kurs II	**79,90**
TI-Writer (deutsch)	**259,–**
Basic Compiler	**98,–**
ID-Data	**69,–**
ID-Konto	**69,–**
Term. Emulator II	**85,–**
Spiele	**ab 20,–**

Alle Preise incl. MwSt. zuzüglich 5,- DM Versandkosten. Lieferung per Nachnahme oder Vorkasse, ab 200,– DM versandkostenfrei.
Fordern Sie unsere kostenlose Preisliste an.

Programm-Service
PSR REIS GmbH
D-5584 Bullay
Bergstraße 80
Telefon 06542/2715

SONDERHEFT
Nr. 4/86/April

INHALT

UTILITIES
SCHULE
ANWENDER
SPIELE
IN TI BASIC
IN EX BASIC

Jungler
Ab Seite 5

Burgenschlacht
Ab Seite 10

Fraggles
Ab Seite 13

Hausarbeiten
Ab Seite 17

Samurai
Ab Seite 21

Verflixte Börse
Ab Seite 31

Apollo
Ab Seite 35

Mission Zulu
Ab Seite 41

Char
Ab Seite 45

Alle Neune
Ab Seite 48

TI-Pittfall
Ab Seite 55

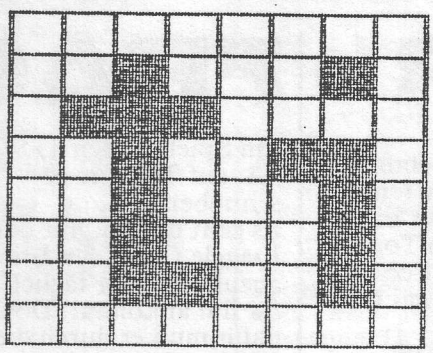
Flugsimulator
Ab Seite 59

Zauberschloß
Ab Seite 66

Adventure III
Ab Seite 86

Irvel II
Ab Seite 91

Superhirn
Ab Seite 83



Grüß Gott

Hier ist sie, nach den beiden Heften im letzten Jahr: Die erste Spezialausgabe für TI-Freaks im Jahr 1986. Rechtzeitig zu den Osterfeiertagen, damit die Zeit nicht zu lang wird. Monatelang haben wieder unsere Programm-Redakteure die von den TI-Lesern eingeschickten Programme gesichtet, überprüft und, wo notwendig, verfeinert. Herausgekommen ist ein Heft voller Super-Listings, das in keiner TI-Bibliothek fehlen darf. Besonderes Augenmerk legten wir dabei auf die sogenannten „Adventures" sowie Spiele gegen den Computer. Ein „error"freies Eintippen wünscht Ihnen Ihr TI-REVUE-Team.

Bis zum nächsten Heft

Let Jump
Ab Seite 96

E.T.
Ab Seite 101

Rettet die Robben
Ab Seite 106

Tank-Commander
Ab Seite 111

Snake
Ab Seite 116

Zahlenjagd
Ab Seite 118

Diadem
Ab Seite 121

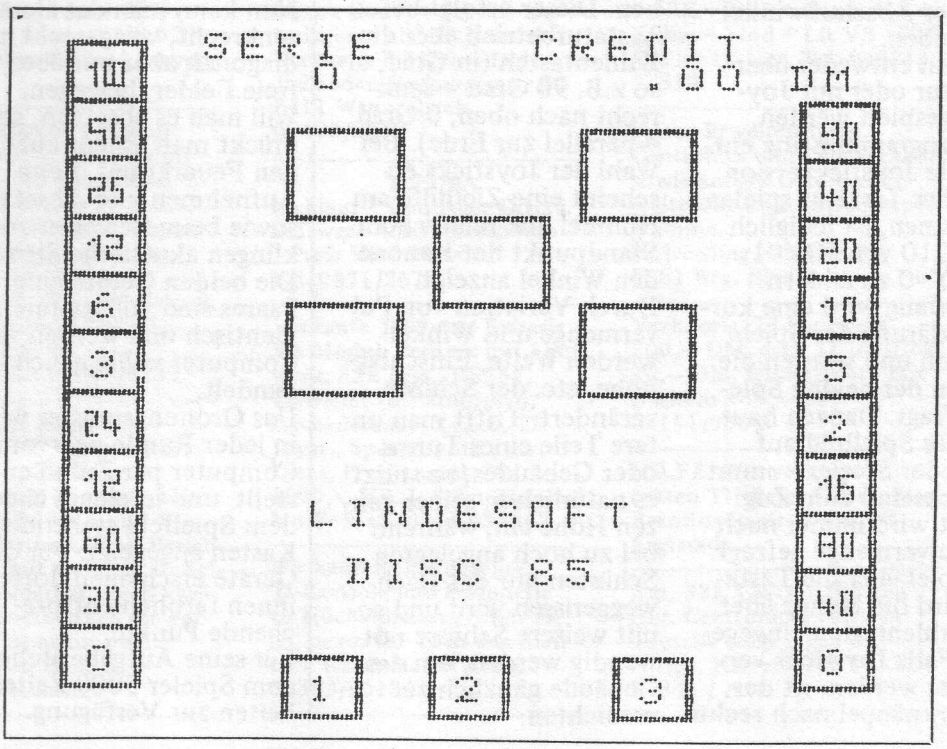
Geldspielautomat
Ab Seite 127

War Games
Ab Seite 134

Long live the King
Ab Seite 136

Pogo Jump
Ab Seite 138

Kreuzworträtsel
Ab Seite 143



JUNGLER

Dieses Spiel ist eine TI-Basicversion von Jungle-Hunt.
Sie müssen Ihre Freundin Jane aus der Hand der Eingeborenen befreien. Das Spiel geht über drei Bilder. Im ersten Bild müssen Sie von Liane zu Liane springen. Im zweiten Bild sind Steine zu überspringen. Beide Aufgaben erledigen Sie entweder mit dem Feuerknopf des Joysticks oder einer beliebigen Taste der linken Tastaturseite. Die Sprünge erfolgen immer nach links. Im dritten Bild haben Sie vier Eingeborene und einen Topf zu überspringen. Folgende Bewegungen sind möglich:
Joystick: Links bzw. rechts gehen mit dem Joystick, soll gesprungen werden ist zusätzlich die Feuertaste zu drücken.
Tastatur: Links gehen (S), rechts gehen (D), links springen (<), rechts springen (>), ohne die FNCT-Taste.

BURGEN-SCHLACHT

Das Spiel für zwei Personen hat eine sorgfältig gestaltete Grafik. Jeder der beiden Spieler schlüpft in die Rolle eines Ritters, der die Aufgabe hat, von seiner Burg aus vermittels einer Kanone die Burg des Kontrahenten in Grund und Boden zu schießen. Hierfür ist die Kanone mit einer zu bestimmenden Menge Schwarzpulver zu laden und der Abschußwinkel anzugeben.
Es kann entweder über Tastatur oder mit Joystick gespielt werden.
Das Programmlisting enthält die Joystickversion. Um über Tastatur spielen zu können, ist lediglich Zeile 210 von JOY=1 in JOY=0 zu ändern.
Zu Anfang wird eine kurze Erklärung des Spiels gegeben und werden die Namen der beiden Spieler erfragt. Danach baut sich das Spielbild auf und jeder Spieler kommt abwechselnd zum Zug. Zuerst wird immer nach der Pulvermenge gefragt. Bei Spiel über die Tastatur wird die Menge über die Zahlentasten eingegeben. Falls Joysticks verwendet werden, ist der Steuerknüppel nach rechts zu ziehen, um die Pulvermenge zu erhöhen, nach links, um sie zu verringern. Die Eingabe wird mit Druck auf die Feuertaste (ENTER bei Tastaturbetrieb) beendet (zuviel Pulver ist gefährlich!).
Negative Zahlen werden am Bildschirm zwar angezeigt, jedoch bei Druck auf den Feuerknopf nicht akzeptiert. Danach ist der Schußwinkel einzugeben. Dieser erfolgt bei Tastaturbetrieb über die Zahlentasten (in Grad, also z.B. 90 Grad = senkrecht nach oben, 0 Grad = parallel zur Erde). Bei Wahl der Joysticks erscheint eine Zielhilfe am Himmel, die relativ zum Standpunkt der Kanone den Winkel anzeigt.
Durch Variation von Pulvermenge und Winkel werden Weite, Einschlaghöhe, etc. der Schüsse verändert. Trifft man untere Teile eines Turms oder Gebäudes, so stürzt es natürlich in seiner ganzen Höhe ein, während bei zu hoch angelegten Schüssen nur das Dach weggerissen wird und somit weitere Schüsse notwendig werden, um das Gebäude gänzlich zu vernichten.

FRAGGLES

Ein Spiel für den TI 99/4A, auf TI-Basic geschrieben.
Es geht darum, mit dem Fraggle Gobo den Brief abzuholen, der täglich für ihn ankommt. Doch dafür muß er durch die Höhle gehen und am Hund Sprocket vorbeikommen.
In der Höhle muß er über Löcher und Hügel springen sowie die Spinnen und Monster passieren.
Die Steuerung erfolgt über Tastatur:
S – Gobo geht einen Schritt nach links.
E – Gobo springt nach links
X – Gobo bückt sich.
Beim Gehen, Springen und Stehenbleiben müssen Sie aufpassen, daß die Spinnen oder das Monster Gobo nicht erwischen, denn dann ist das Spiel zu Ende. Beim Bücken dürfen Sie Gobo nicht mit dem Kopf auf einen Berg stoßen, sonst ist das Spiel ebenfalls zu Ende. Doch die gebückte Haltung hat den Vorteil, daß das Monster und die Spinnen Gobo nicht berühren können.
Nach dem Spiel wird Ihre erreichte Punktzahl angezeigt.

HAUSARBEITEN

Die Aufgabe des Spielers ist es, zehn Haushaltsgeräte, die in fünf Paare aufgeteilt sind, in einer vorgegebenen Ordnung auf einem quadratischen Brett mit 25 Felder anzuordnen.
Der Joystick lenkt dabei ein Kreuz, mit dessen Hilfe man die Geräte verschieben kann. Zu diesem Zweck geht man mit dem Kreuz auf ein Gerät und drückt den Feuerknopf. Nun kann man das Gerät senkrecht, waagerecht und diagonal, aber nur über freie Felder, bewegen. Will man es absetzen, so drückt man erneut auf den Feuerknopf. Beim Aufnehmen und Absetzen sowie beim Bewegen erklingen akustische Signale.
Die beiden Geräte eines Paares sind vollkommen identisch und werden vom Computer völlig gleich behandelt.
Das Ordnungsmuster wird in jeder Runde neu vom Computer per Zufall erstellt und in einem über dem Spielfeld stehenden Kasten angezeigt. Für die Geräte erscheinen dort ihnen farblich entsprechende Punkte.
Für seine Aufgabe stehen dem Spieler 5000 Zeiteinheiten zur Verfügung. Schafft er es innerhalb dieser Zeit nicht, die vorgegebene Ordnung herzustellen, so wird ihm einer seiner insgesamt vier Versuche abgezogen, und er spielt bei dem augenblicklichen Stand mit 5000 neuen Einheiten weiter.
Nach dem vierten vergeblichen Versuch ist das Spiel zu Ende.
Um dem Spieler seine Aufgabe nicht allzu leicht zu machen, sind drei Schwierigkeiten eingebaut. Zum einen werden dem Spieler beim Absetzen eines Gerätes 50 Zeiteinheiten abgezogen, falls der Zufall es so will. Die Möglichkeit dieses Zufalls steigt von Runde zu Runde.
Setzt man ein Gerät auf einem Feld ab, für das es nicht vorgesehen ist, so erscheint eine Hand, die irgendein Gerät aufnimmt und woanders absetzt, so daß man häufig gezwungen ist, dieses Gerät unter Zeitverlust wieder umzusetzen.
Die dritte Schwierigkeit besteht darin, daß vom Computer gesetzte Blökke freie Felder belegen und somit den Platz

Bitte lesen Sie weiter auf Seite 9

LISTINGS

```
10 REM*********************
11 REM*      JUNGLER       *
12 REM*                    *
13 REM*   Copyright by     *
14 REM*                    *
15 REM*  Christoph Paul    *
16 REM*                    *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM* (evtl. Joystick) *
20 REM*                    *
23 REM* Speicherbelegung *
24 REM*   10499 Bytes     *
25 REM*********************
26 REM
100 HIGHSCORE=5000
110 DIM KA(5)
120 CALL CLEAR
130 CALL COLOR(1,16,16)
140 FOR I=2 TO 14
150 CALL COLOR(I,16,16)
160 NEXT I
170 CALL SCREEN(2)
180 DATA 0303030303030303,03
0303030303FF7E,C3C3C3C3C3C3C
3C3,C3C3C3C3C3C3FF7E,C3C3E3E
3E3D3D3D3
190 DATA CBCBCBC7C7C7C3C3,7E
FFC0C0C0C0C0C0,CECFC3C3C3C3F
F7E,C0C0C0C0C0C0C0C0,C0C0C0C
0C0C0FF7E
200 DATA 7EFFC0C0C0C0C0FE,FE
C0C0C0C0C0FF7E,7EFFC3C3C3C3F
FFE,F0F0D8D8CCCCC6C6
210 RESTORE 180
220 FOR I=97 TO 110
230 READ A$
240 CALL CHAR(I,A$)
250 NEXT I
260 CALL CHAR(64,"3C4299A1A1
99423C")
270 FOR I=1 TO 24
280 PRINT
290 NEXT I
300 PRINT ".......a c e g i
k m..............b d f h j
l n": : : : : : : :".....@ 1
985 BY DATA CP": :".....HIGH
SCORE:";
310 PRINT SEG$("0000000",1,7
-LEN(STR$(HIGHSCORE)))&STR$(
HIGHSCORE)
320 PRINT : : : :"...PRESS A
NY KEY TO START": :
330 FOR I=2 TO 14
340 CALL COLOR(I,2,16)
350 NEXT I
360 DATA 100,262,100,330,100
,349,200,392,100,40000,100,2
62,100,330,100,349,200,392,1
00,40000
370 DATA 100,262,100,330,100
,349,200,392,150,330,160,294
,149,262,349,262,999,42000
380 RESTORE 360
390 FOR I=1 TO 19
400 READ A,B
410 CALL SOUND(A+85,B,0,B+9,
19)
420 NEXT I
430 CALL KEY(1,K,S)
440 CALL KEY(0,K,ST)
450 IF (S=0)*(ST=0)THEN 430
460 CALL VCHAR(1,3,32,672)
470 DATA 007C44444464647C,00
1010101018181 8,007C04047C606
07C,007C04041C0C0C7C,0040444
47C0C0C0C
480 DATA 007C40407C0C0C7C,00
7C40407C64647C,007C0404040C0
C0C,007C44447C64647C,007C444
47C0C0C7C
490 RESTORE 470
500 FOR I=48 TO 57
510 READ A$
520 CALL CHAR(I,A$)
530 NEXT I
540 DATA 40,181866BD99242442
,41,181A6FBD9A242412,42,1858
F6BD59242448,43,08142A1C2A08
1422,96,98A498FE99BCA4A4
550 DATA 97,7EC3FF7E3CBC522C
,104,3C7AFDFFFFFF7E3C,105,1C
7EE7011D7FE701,106,387EE780B
8FEE78,112,0101010101010101
560 DATA 98,FFFFFFFFFFFFFFFF
,99,183C7EFF5A567A76,118,010
3070F1F3F7FFF,119,446655AA55
AA55AA
570 DATA 113,8080808080808080
,114,FFFF1F0701010301,115,7F
1F,116,0103030101010301,117,
FFFFFFFFFFFFFFFF
580 RESTORE 540
590 FOR I=1 TO 19
600 READ A,A$
610 CALL CHAR(A,A$)
620 NEXT I
630 DATA 2,5,16,3,16,2,4,16,
2,5,16,2,6,16,2,7,16,2,8,16,
2,9,2,16,10,3,16,11,9,16
640 RESTORE 630
650 FOR I=1 TO 10
660 READ C1,C2,C3
670 CALL COLOR(C1,C2,C3)
680 NEXT I
690 FOR I=4 TO 21
700 CALL HCHAR(I,3,32,28)
710 NEXT I
720 ZAEHL=0
730 FOR I=1 TO 3
740 CALL HCHAR(I,3,98,28)
```

LISTINGS

```
750 CALL HCHAR(I+21,3,98,28)
760 NEXT I
770 MAN=50
780 SCORE=0
790 LEVEL=1
800 GOTO 850
810 FOR II=1 TO LEN(A$)
820 CALL HCHAR(RE,(ZE-1)+II,
ASC(SEG$(A$,II,1)))
830 NEXT II
840 RETURN
850 A$="JUNGLERb@b1985bBYbDA
TAbCP"
860 RE=2
870 ZE=4
880 GOSUB 810
890 A$="SCORE:bbbbbbbbbbbbbJ
UNGLER:b"
900 RE=23
910 ZE=3
920 GOSUB 810
930 CALL HCHAR(23,30,(48+MAN
))
940 A$=STR$(SCORE)
950 A$=SEG$("00000000",1,7-LE
N(A$))&A$
960 RE=23
970 ZE=9
980 GOSUB 810
990 CALL HCHAR(2,3,98,28)
1000 FOR I=27 TO 30
1010 CALL VCHAR(4,I,117,18)
1020 NEXT I
1030 CALL VCHAR(4,26,116,18)
1040 CALL HCHAR(4,25,115)
1050 CALL HCHAR(4,26,114)
1060 CALL HCHAR(12,25,115)
1070 CALL HCHAR(12,26,114)
1080 CALL HCHAR(21,3,119,28)
1090 FOR I=4 TO 22 STEP 3
1100 CALL HCHAR(19,I,105)
1110 CALL HCHAR(19,I+1,106)
1120 CALL HCHAR(20,I,112)
1130 CALL HCHAR(20,I+1,113)
1140 NEXT I
1150 CALL HCHAR(20,25,118)
1160 CALL HCHAR(19,26,118)
1170 CALL HCHAR(20,26,117)
1180 CALL VCHAR(4,4,113,9)
1190 A=11
1200 B=25
1210 @1=+1
1220 @2=-1
1230 L1=10
1240 L2=20
1250 CALL HCHAR(A,B,40)
1260 CALL VCHAR(4,L1,113,9)
1270 CALL VCHAR(4,L2,113,9)
1280 GOTO 1510
1290 CALL VCHAR(4,L1,32,9)
1300 CALL VCHAR(4,L2,32,9)
1310 L1=L1+@1
1320 L2=L2+@2
1330 IF L1>12 THEN 1340 ELSE
 1350
1340 @1=-1
1350 IF L1<8 THEN 1360 ELSE
1370
1360 @1=+1
1370 IF L2>22 THEN 1380 ELSE
 1390
1380 @2=-1
1390 IF L2<18 THEN 1400 ELSE
 1410
1400 @2=+1
1410 CALL HCHAR(A,B,32)
1420 CALL VCHAR(4,L1,113,9)
1430 CALL VCHAR(4,L2,113,9)
1440 IF (B<25)*(B>16)THEN 14
50 ELSE 1470
1450 B=L2
1460 GOTO 1490
1470 IF (B<14)*(B>6)THEN 148
0 ELSE 1490
1480 B=L1
1490 CALL HCHAR(A,B,41)
1500 RETURN
1510 GOSUB 1290
1520 IF B<5 THEN 2300
1530 CALL KEY(1,KEY,STAT)
1540 CALL SOUND(-555,-8,2)
1550 IF STAT=0 THEN 1510
1560 CALL HCHAR(A,B,32)
1570 A=A-1
1580 B=B-1
1590 CALL GCHAR(A,B,C)
1600 IF C=113 THEN 2200
1610 CALL HCHAR(A,B,41)
1620 CALL HCHAR(A,B,32)
1630 B=B-1
1640 CALL GCHAR(A,B,C)
1650 IF C=113 THEN 2200
1660 CALL HCHAR(A,B,43)
1670 CALL HCHAR(A,B,32)
1680 A=A+1
1690 B=B-1
1700 CALL GCHAR(A,B,C)
1710 IF C=113 THEN 2200
1720 CALL HCHAR(A,B,43)
1730 CALL HCHAR(A,B,32)
1740 A=A+1
1750 B=B-1
1760 CALL GCHAR(A,B,C)
1770 IF C=113 THEN 2200
1780 CALL HCHAR(A,B,43)
1790 CALL HCHAR(A,B,32)
1800 A=A+1
1810 IF A>18 THEN 1880
1820 GOSUB 1290
1830 CALL GCHAR(A,B,C)
1840 IF C=113 THEN 2200
1850 CALL HCHAR(A,B,40)
```

LISTINGS

```
1860 CALL HCHAR(A,B,32)
1870 GOTO 1800
1880 REM KAPUTT
1890 MAN=MAN-1
1900 FOR I=1 TO 30 STEP 2
1910 CALL SOUND(55,I+110,I)
1920 NEXT I
1930 CALL SOUND(-999,-5,0)
1940 CALL HCHAR(23,30,(48+MA
N))
1950 CALL VCHAR(4,L1,32,9)
1960 CALL VCHAR(4,L2,32,9)
1970 IF MAN>0 THEN 1190
1980 A$="GbAbMbEbbbObVbEbR"
1990 RE=12
2000 ZE=8
2010 GOSUB 810
2020 A$="LEVEL:"&STR$(LEVEL)
2030 RE=2
2040 ZE=13
2050 GOSUB 810
2060 IF SCORE>=HIGHSCORE THE
N 2070 ELSE 2080
2070 HIGHSCORE=SCORE
2080 FOR @@=1 TO 1000
2090 NEXT @@
2100 A$="bbbbbPRESSbREDObORb
BACKbbbbb"
2110 RE=2
2120 ZE=3
2130 GOSUB 810
2140 CALL KEY(0,K,S)
2150 IF K=6 THEN 690
2160 IF K=15 THEN 120
2170 IF S=0 THEN 2140
2180 CALL SOUND(200,220,0)
2190 GOTO 2140
2200 REM
2210 CALL HCHAR(A,B,41)
2220 SCORE=SCORE+100
2230 A$=STR$(SCORE)
2240 A$=SEG$("0000000",1,7-L
EN(A$))&A$
2250 RE=23
2260 ZE=9
2270 GOSUB 810
2280 IF B<5 THEN 2300
2290 GOTO 1510
2300 ZAEHL=ZAEHL+1
2310 IF ZAEHL>4 THEN 2350
2320 CALL VCHAR(4,L1,32,9)
2330 CALL VCHAR(4,L2,32,9)
2340 GOTO 1180
2350 ZAEHL=0
2360 FOR I=4 TO 21
2370 CALL HCHAR(I,3,32,28)
2380 NEXT I
2390 CALL HCHAR(21,3,119,28)
2400 FOR I=4 TO 28 STEP 3
2410 CALL HCHAR(15,I,105)
2420 CALL HCHAR(15,I+1,106)
2430 CALL HCHAR(16,I,112)
2440 CALL HCHAR(16,I+1,113)
2450 NEXT I
2460 A=20
2470 B=29
2480 C=0
2490 @=4
2500 CALL HCHAR(A,B,40)
2510 CALL HCHAR(20,@,104)
2520 C=C+.5
2530 IF C=INT(C)THEN 2540 EL
SE 2560
2540 CALL HCHAR(A,B,32)
2550 B=B-1
2560 IF B<8 THEN 3010
2570 CALL HCHAR(20,@,32)
2580 @=@+1
2590 CALL HCHAR(A,B,41)
2600 CALL KEY(1,K,S)
2610 IF S<>0 THEN 2710
2620 CALL HCHAR(20,@,104)
2630 IF @=B THEN 2900
2640 IF @>B THEN 2670
2650 CALL SOUND(95,440,9,-2,
4)
2660 GOTO 2560
2670 CALL SOUND(100,-3,8)
2680 CALL HCHAR(20,@,32)
2690 GOTO 2830
2700 GOTO 2490
2710 CALL HCHAR(A,B,32)
2720 A=A-1
2730 B=B-1
2740 CALL HCHAR(A,B,41)
2750 CALL HCHAR(20,@,32)
2760 @=@+1
2770 CALL HCHAR(20,@,104)
2780 CALL HCHAR(A,B,32)
2790 A=A+1
2800 B=B-1
2810 CALL HCHAR(A,B,41)
2820 GOTO 2620
2830 SCORE=SCORE+50
2840 A$=STR$(SCORE)
2850 A$=SEG$("0000000",1,7-L
EN(A$))&A$
2860 RE=23
2870 ZE=9
2880 GOSUB 810
2890 GOTO 2700
2900 REM
2910 FOR I=1 TO 30 STEP 2
2920 CALL SOUND(55,I+110,I)
2930 NEXT I
2940 CALL SOUND(-999,-5,0)
2950 MAN=MAN-1
2960 CALL HCHAR(23,30,(48+MA
N))
2970 CALL HCHAR(A,B,32)
2980 CALL HCHAR(20,@,32)
2990 IF MAN<1 THEN 1980
```

LISTINGS

```
3000 GOTO 2460
3010 CALL HCHAR(A,B,32)
3020 CALL HCHAR(20,@,32)
3030 SCORE=SCORE+100
3040 ZAEHL=ZAEHL+1
3050 IF ZAEHL<5 THEN 2460
3060 ZAEHL=0
3070 REM
3080 FOR I=4 TO 20
3090 CALL HCHAR(I,3,32,28)
3100 NEXT I
3110 FOR I=7 TO 30
3120 CALL HCHAR(INT(RND*5)+1
0,I,99)
3130 NEXT I
3140 FOR I=4 TO 29 STEP 3
3150 CALL HCHAR(14,I,105)
3160 CALL HCHAR(14,I+1,106)
3170 CALL HCHAR(15,I,112)
3180 CALL HCHAR(15,I+1,113)
3190 NEXT I
3200 CALL VCHAR(18,3,113,3)
3210 CALL HCHAR(17,3,106)
3220 CALL HCHAR(20,4,97)
3230 CALL HCHAR(17,4,113)
3240 CALL HCHAR(18,4,43)
3250 A=20
3260 B=29
3270 KA(1)=9
3280 KA(2)=13
3290 KA(3)=17
3300 KA(4)=21
3310 KA(5)=25
3320 @=+1
3330 CALL HCHAR(A,B,40)
3340 FOR I=1 TO 5
3350 CALL HCHAR(20,KA(I),96)
3360 NEXT I
3370 GOTO 3500
3380 FOR I=1 TO 5
3390 CALL HCHAR(20,KA(I),32)
3400 NEXT I
3410 IF KA(1)>11 THEN 3420 E
LSE 3430
3420 @=-1
3430 IF KA(1)<7 THEN 3440 EL
SE 3450
3440 @=+1
3450 FOR I=1 TO 5
3460 KA(I)=KA(I)+@
3470 CALL HCHAR(20,KA(I),96)
3480 NEXT I
3490 RETURN
3500 CALL JOYST(1,X,Y)
3510 CALL KEY(1,K,S)
3520 IF (X=4)*(K=18)THEN 383
0
3530 IF (X=-4)*(K=18)THEN 39
50
3540 IF X=4 THEN 3670
3550 IF X=-4 THEN 3750
3560 CALL KEY(0,K,S)
3570 IF K=ASC("D")THEN 3670
3580 IF K=ASC("S")THEN 3750
3590 IF K=ASC(",")THEN 3950
3600 IF K=ASC(".")THEN 3830
3610 GOTO 4080
3620 GOSUB 3380
3630 FOR I=1 TO 5
3640 IF B=KA(I)THEN 4120
3650 NEXT I
3660 GOTO 3500
3670 CALL HCHAR(A,B,32)
3680 B=B+1
3690 IF B>30 THEN 3760
3700 FOR I=1 TO 5
3710 IF B=KA(I)THEN 4120
3720 NEXT I
3730 CALL HCHAR(A,B,42)
3740 GOTO 3620
3750 CALL HCHAR(A,B,32)
3760 B=B-1
3770 IF B<5 THEN 4120
3780 FOR I=1 TO 5
3790 IF B=KA(I)THEN 4120
3800 NEXT I
3810 CALL HCHAR(A,B,41)
3820 GOTO 3620
3830 IF (B>28)THEN 3610
3840 CALL HCHAR(A,B,32)
3850 A=A-1
3860 B=B+1
3870 CALL HCHAR(A,B,42)
3880 IF (A=19)*(B=4)THEN 425
0
3890 GOSUB 3380
3900 CALL HCHAR(A,B,32)
3910 A=A+1
3920 B=B+1
3930 CALL HCHAR(A,B,40)
3940 GOTO 4080
3950 IF (B=5)THEN 3970
3960 IF (B<7)THEN 3610
3970 CALL HCHAR(A,B,32)
3980 A=A-1
3990 B=B-1
4000 CALL HCHAR(A,B,41)
4010 GOSUB 3380
4020 IF (A=19)*(B=4)THEN 425
0
4030 CALL HCHAR(A,B,32)
4040 A=A+1
4050 B=B-1
4060 CALL HCHAR(A,B,40)
4070 GOTO 4080
4080 FOR I=1 TO 5
4090 IF B=KA(I)THEN 4120
4100 NEXT I
4110 GOTO 3620
4120 REM
4130 MAN=MAN-1
4140 FOR I=1 TO 30 STEP 2
```

```
4150 CALL SOUND(55,I+110,I)
4160 NEXT I
4170 CALL SOUND(-999,-5,0)
4180 CALL HCHAR(23,30,(48+MA
N))
4190 CALL HCHAR(A,B,32)
4200 IF MAN<1 THEN 1980
4210 FOR I=1 TO 5
4220 CALL HCHAR(20,KA(I),32)
4230 NEXT I
4240 GOTO 3250
4250 FOR I=1 TO 20
4260 CALL SOUND(I+55,I*110,0
,I+110,10,I+I+110,20)
4270 NEXT I
4280 GOTO 4370
4290 SCORE=SCORE+1000
4300 A$=STR$(SCORE)
4310 A$=SEG$("00000000",1,7-L
EN(A$))&A$
4320 RE=23
4330 ZE=9
4340 GOSUB 810
4350 LEVEL=LEVEL+1
4360 RETURN
4370 FOR I=4 TO 21
4380 CALL HCHAR(I,3,32,28)
4390 NEXT I
4400 DATA 5,5,4,8,5,3,9,7,2,
7,7,2,11,5,4,5,11,2,11,11,2,
11,23,2,5,27,3,8,27,3,11,27,
3
4410 DATA 14,12,4,19,12,4,14
,17,4,19,17,4,14,22,4,19,22,
4
4420 RESTORE 4400
4430 CALL HCHAR(6,8,98)
4440 CALL HCHAR(10,8,98)
4450 FOR I=13 TO 23 STEP 5
4460 CALL VCHAR(17,I,98,2)
4470 NEXT I
4480 FOR I=1 TO 17
4490 READ A1,A2,A3
4500 CALL HCHAR(A1,A2,98,A3)
4510 NEXT I
4520 DATA 5,5,7,6,10,5,6,13,
5,5,15,7,5,20,7,6,16,2,7,17,
2,8,18,2,9,19,2,5,22,6,5,25,
6
4530 DATA 5,27,4,8,29,4,14,8
,6,17,9,3,14,12,6,14,15,6,14
,17,6,14,20,6,14,22,6,14,25,
6
4540 RESTORE 4520
4550 FOR I=1 TO 21
4560 READ A1,A2,A3
4570 CALL VCHAR(A1,A2,98,A3)
4580 NEXT I
4590 GOSUB 4290
4600 FOR I=16 TO 2 STEP -1
4610 CALL COLOR(9,I,16)
4620 CALL SOUND(99,I*110,5)
4630 NEXT I
4640 FOR I=4 TO 21
4650 CALL HCHAR(I,3,32,28)
4660 NEXT I
4670 GOTO 850
```

LISTINGS

Fortsetzung von Seite 4

HAUSARBEITEN

zum Rangieren beschränken. In der ersten Runde wird nur ein Block gesetzt. Jede zweite Runde folgt ein weiterer, bis in der siebten vier Blöcke vorhanden sind. Die Anzahl der Blöcke ist auf vier beschränkt, da das Spiel sonst zu unübersichtlich würde. Im Notfall können die Blöcke nach dem oben beschriebenen Prinzip versetzt werden, was jedoch 500 Zeiteinheiten kostet, und deshalb nicht unbedingt zu empfehlen ist.

Eine weitere Schwierigkeit baut der TI selbst ein. Auf jeder Zeile können nur vier Sprites dargestellt werden. Der fünfte ist nicht mehr sichtbar und kann auch nicht aufgenommen und versetzt werden. Zu diesem Zweck muß zuvor ein anderes Gerät aus dieser Zeile entfernt werden. Man sollte sich also genau überlegen, in welcher Reihenfolge man die Geräte auf ihren Feldern absetzt.

SAMURAI

Sie sind ein japanischer Strauchritter (Ronin), der gegen Samurais, Kyodoka und Vögel kämpfen muß, um am Ende vom obersten Herrscher der Provinz (Daimyo) zum Samurai ernannt und in seine Dienste aufgenommen zu werden. Damit Sie die gute Grafik betrachten können, bleiben die Szenen auch nach erfolgreichem Kampf noch bestehen, bis eine Taste gedrückt wird. Dasselbe gilt für eingefügte erklärende Texte.

In der ersten Szene müssen die heranstürmenden Samurai durch einen Schwertstoß außer Gefecht gesetzt werden. Steuerung: Joystick links (rechts): Samurai läuft links (rechts); Schwertstoß: Joystick nach unten; Sprung über den Gegner: Joystick nach oben.

In der zweiten Szene steht Ihnen nur ein Gegner gegenüber. Rempeln Sie ihn von hinten (Joystick links + Feuertaste)! Tun Sie dies nicht, so weicht er Ihren Schwertstößen aus. Erst nach dem Rempeln können Sie ihn besiegen.

In der dritten Szene müssen Sie gegen Bogenschützen kämpfen. Weichen Sie den Pfeilen aus, erledigen Sie den Schützen und holen Sie den Vogel herunter (Joystick nach unten), um ihm den Schlüssel zur Burg abzunehmen. Springen Sie den Vogel nicht in der Luft an! Es braucht etwas Übung, um den Vogel in die richtige Position zu bringen, um ihm den Schlüssel wegzunehmen. Wenn Ihr Versuch erfolgreich ist, ertönt ein kurzer, hoher Ton.

Nachdem Sie den folgenden Text mit einem Tastendruck quittiert haben, gelangen Sie vor den Daimyo.

Zur Programmeingabe:
Das Programm ist in zwei Teile gegliedert, so daß eine Speichererweiterung nicht benötigt wird. Der zweite Teil wird automatisch nachgeladen. Diskettenbesitzer müssen den RUN "CS1"-Befehl umschreiben und eventuell vorher CALL FILES(1) und NEW eingeben!

Bitte lesen Sie weiter auf Seite 13

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *   BURGENSCHLACHT     *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Juergen Krause     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *  (opt. Joysticks)    *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     7794 Bytes       *
28 ! *                      *
29 ! ************************
100 GOTO 150 !SCHNELLSTART
110 CALL LOCATE :: CALL CHAR
SET :: CALL CLEAR :: CALL VC
HAR :: CALL SOUND :: CALL CO
LOR :: CALL SCREEN :: CALL K
EY :: CALL JOYST
120 CALL SPRITE :: CALL CHAR
 :: CALL DELSPRITE(ALL):: CA
LL WIND :: CALL SCHUSS :: CA
LL TAST :: CALL JOY :: CALL
TITEL :: CALL BILD
130 G,VY,VX,FI,V,D1,T1,L1,T2
,L2,JOY,NA1$,NA2$
140 CALL POSITION :: CALL MO
TION :: CALL GCHAR :: CALL H
CHAR :: CALL PATTERN
150 !@P-
160 RANDOMIZE
170 CALL SCREEN(8)
180 CALL DELSPRITE(ALL)
190 CALL CHARSET
200 ON WARNING NEXT
210 JOY=0 !JOY=1:JOYSTICK
            JOY=0:TASTATUR
220 CALL CLEAR
230 CALL TITEL(NA1$,NA2$,JOY
)
240 CALL BILD
250 D1=60 :: T1=110 :: L1=7
:: T2=-5 :: L2=0 !ABSCHUSSTO
N
260 !HAUPTPROGRAMM
270 CALL DELSPRITE(#1):: IF
JOY=1 THEN CALL JOY(1,VX,VY,
V,FI,NA1$)ELSE CALL TAST(1,V
X,VY,V,FI,NA1$)
280 CALL SPRITE(#1,118,10,92
,65):: CALL SOUND(D1,T1,L1,T
2,L2)
290 IF FI>PI/2 OR FI<0 OR V>
23 OR VX<0 THEN CALL ENDE("R
ABSTEIN",NA2$)
300 CALL SCHUSS(1,VX,VY,FI)
310 CALL GCHAR(13,25,G):: IF
 G=99 THEN CALL HCHAR(13,25,
41):: CALL SPRITE(#21,110,2,
97,193)
320 CALL GCHAR(13,32,G):: IF
 G=99 THEN CALL ENDE("WALLBE
RG",NA1$)
330 CALL DELSPRITE(#1):: IF
JOY=1 THEN CALL JOY(2,VX,VY,
V,FI,NA2$)ELSE CALL TAST(2,V
X,VY,V,FI,NA2$)
340 CALL SPRITE(#1,117,10,92
,185):: CALL SOUND(D1,T1,L1,
T2,L2)
350 IF FI>PI/2 OR FI<0 OR V>
23 OR VX<0 THEN CALL ENDE("W
ALLBERG",NA1$)
360 CALL SCHUSS(2,VX,VY,FI)
370 CALL GCHAR(13,8,G):: IF
G=99 THEN CALL HCHAR(13,8,41
):: CALL SPRITE(#20,109,2,97
,57)
380 CALL GCHAR(13,1,G):: IF
G=99 THEN CALL ENDE("RABSTEI
N",NA2$)
390 GOTO 270
400 !@P+
410 SUB BILD
420 !HINTERGRUND WEISS
430 CALL SCREEN(16)
440 ! BUCHSTABEN DUNKELROT
450 FOR I=3 TO 8 :: CALL COL
OR(I,7,1):: NEXT I
460 ! SONDERZEICHEN
470 A$="3333"&RPT$("0",12)
480 B$="00101010"&RPT$("0",8
)
490 C$="8080808080808080"
500 D$="8090909080808080"
510 E$="0103070F1F3F7FFF"
520 F$="80C0E0F0F8FCFEFF"
530 G$="FFFFFFFF00000000"
540 H$="E7C3810000000000"
550 I$="7F2F0B0303030100"
560 J$="FEFAF0F0F0E08000"
570 K$="7F3F0501000000000"
580 L$="FFFFF7F020000000"
590 M$="FEF0E0A000000000"
600 N$="CCCC000000000000"
610 O$="00000000183C7EFF"
620 HX$="183C7EFFFFFFFFFF"
630 FX$="7F3F1F0F07030100"
640 EX$="FEFCF8F0E0C08000"
650 !@P- FARBEN
660 CALL COLOR(2,8,4,9,8,16)
670 CALL COLOR(10,2,16,11,7,
8,12,3,4)
680 CALL COLOR(13,8,3,14,4,1
5,0,2,8)
690 ! ZEICHENDEFINITION
700 CALL CHAR(40,I$&RPT$("F"
,16),91,G$&"C0C0"&RPT$("0",2
```

LISTINGS

```
8),96,A$&N$&RPT$("0",16)&RPT
$("F",16))
710 CALL CHAR(100,"80C0E0E0F
87C3C0C8080C0C0C0E0602001010
303030706040103070 71F3E3C30"
)
720 CALL CHAR(104,B$&C$&D$&H
X$)
730 CALL CHAR(108,RPT$("F",1
6)&RPT$("0",8)&"030C3ECE"&RP
T$("0",8)&"C0307C73")
740 CALL CHAR(112,E$&F$&O$&R
PT$("F",16),116,C$)
750 CALL CHAR(117,"40289A7C3
EFE1F030214593E7C7FF8C010927
C3FBC783810")
760 CALL CHAR(120,I$&K$&L$&M
$)
770 CALL CHAR(124,J$&RPT$("0
",16))
780 CALL CHAR(128,J$&K$&L$&I
$)
790 CALL CHAR(132,RPT$("0",1
6)&M$)
800 CALL CHAR(136,RPT$("0",1
6)&F$&FX$&E$)
810 CALL CHAR(140,EX$)
820 !@P+ ZEICHENPOSITIONEN
830 DATA 99,32,99,32,99,32,9
9,32
840 !@P-
850 DATA 99,32,99,3,116,1,99
,26,116,1,99,1
860 DATA 97,1,99,2,116,1,99,
26,116,1,99,1
870 DATA 98,1,99,1,112,1,113
,1,99,26,96,2
880 DATA 98,1,99,1,104,1,105
,1,99,26,104,1,98,1
890 DATA 98,1,99,1,98,1,106,
1,99,2,114,1,99,11
900 DATA 128,1,132,1,131,1,9
9,3,114,1,99,1,112,1
910 DATA 115,3,98,1,104,1
920 DATA 98,1,115,1,104,1,10
5,1,115,2,104,1,99
930 DATA 10,128,1,132,3,131,
1,99,2,104,1,99,1,104,6
940 DATA 91,6,104,1,113,1,99
,5,128,1,129,1,130
950 DATA 1,133,1,132,5,131,1
,99,1,104,1,91,7
960 DATA 107,1,104,7,99,4,12
8,1,132,10,131,1
970 DATA 104,7,107,1,136,1,1
25,7,40,1,99,2,128,1
980 DATA 132,11,123,1,125,7,
136,1
990 DATA 137,1,138,1,125,7,4
0,1,128,1,132,11
1000 DATA 124,1,125,7,140,1,
139,1
1010 DATA 125,1,137,1,138,1,
125,7,120,1,132,10
1020 DATA 124,1,125,7,140,1,
139,1,125,1
1030 DATA 125,2,137,1,138,1,
125,7,121,1,122,8
1040 DATA 123,1,125,7,140,1,
139,1,125,2
1050 DATA 125,3,137,1,136,24
,139,1,125,3
1060 DATA 125,32,125,32
1070 ! BILDSCHIRMAUFBAU
1080 RESTORE
1090 CALL CLEAR
1100 !@P+
1110 Z,S=1
1120 READ C,A
1130 CALL HCHAR(Z,S,C,A)
1140 S=S+A :: IF S=33 THEN S
=1 :: Z=Z+1
1150 IF Z<>21 THEN 1120
1160 SUBEND
1170 SUB JOY(J,VX,VY,V,FI,NA
$)
1180 GOTO 1210
1190 S,K,X,Y
1200 !@P-
1210 CALL WIND(VX):: DISPLAY
 AT(21,1):NA$
1220 V=1 :: DISPLAY AT(23,1)
:"WIEVIEL KG SCHWARZPULVER?"
1230 CALL KEY(J,K,S):: IF K=
18 THEN 1240 ELSE CALL JOYST
(J,X,Y):: V=V+X/4 :: IF X=0
THEN 1230 ELSE DISPLAY AT(23
,26)BEEP SIZE(3):V :: GOTO 1
230
1240 IF V<1 THEN 1230 ELSE C
ALL SOUND(200,200,0):: CALL
SOUND(10,2000,0):: DISPLAY A
T(23,1):".....ZUENDEN MIT <F
IRE>...." :: V=15+V
1250 CALL SPRITE(#10,42,2,1,
125)
1260 CALL KEY(J,K,S)
1270 CALL JOYST(J,X,Y)
1280 CALL MOTION(#10,0,X)
1290 IF K<>18 THEN 1260
1300 CALL POSITION(#10,Y,X):
: CALL DELSPRITE(#10)
1310 IF J=1 THEN FI=ATN((100
-Y)/(X-65))ELSE FI=ATN((100-
Y)/(190-X))
1320 IF J=1 THEN VX=COS(FI)*
V-VX ELSE VX=COS(FI)*V+VX
1330 VY=SIN(FI)*V
1340 CALL HCHAR(21,1,32,128)
1350 !@P+
1360 SUBEND
1370 SUB TAST(J,VX,VY,V,FI,N
```

```
A$)
1380 !@P-
1390 CALL WIND(VX):: DISPLAY
 AT(21,1):NA$
1400 DISPLAY AT(23,1):"WIEVI
EL KG SCHWARZPULVER?" :: ACC
EPT AT(23,27)SIZE(2)VALIDATE
(DIGIT):V :: V=V+15
1410 DISPLAY AT(23,1):"WELCH
ER WINKEL?" :: ACCEPT AT(23,
25)SIZE(3)VALIDATE(DIGIT):FI
1420 FI=FI/180*PI
1430 IF J=1 THEN VX=COS(FI)*
V-VX ELSE VX=COS(FI)*V+VX
1440 VY=SIN(FI)*V
1450 !@P+
1460 SUBEND
1470 SUB SCHUSS(J,VX,VY,FI)
1480 IF J=1 THEN A=65 :: E=2
56 :: S=3 ELSE A=190 :: E=1
:: S=-3
1490 D1=60 :: T1=110 :: L1=7
 :: T2=-5 :: L2=0 :: G,V,SX,
SY=0 !TREFFERTON
1500 !@P-
1510 CALL PATTERN(#1,92)
1520 VY=2.65*VY :: VX=2.12*V
X
1530 V=(J-1.5)*2
1540 FOR SX=A TO E STEP S
1550 SY=INT(5*(V*(A-SX)/VX)^
2-VY*V*(A-SX)/VX+100)
1560 IF SY>1 THEN CALL LOCAT
E(#1,SY,SX)ELSE CALL LOCATE(
#1,250,250)
1570 IF SY>100 THEN IF SY>12
5 OR V*SX<V*(128-64*V)THEN S
X,SY=1 :: GOTO 1600
1580 IF SY>20 AND V*SX<V*(12
8-64*V)THEN CALL GCHAR(INT((
SY-1)/8+1),INT((SX-1)/8+1),G
):: IF G<>99 THEN 1600
1590 NEXT SX :: SUBEXIT
1600 CALL PATTERN(#1,119)::
CALL SOUND(D1,T1,L1,T2,L2)
1610 CALL VCHAR(1,INT((SX-1)
/8+1),99,MIN(13,INT((SY-1)/8
+1)))
1620 !@P+
1630 SUBEND
1640 SUB ENDE(X$,NA$)
1650 !@P-
1660 DISPLAY AT(21,1):"IN BU
RG ";X$;" IST DIE":"PULVERKA
MMER EXPLODIERT":NA$:"HAT GE
WONNEN" :: RUN
1670 !@P+
1680 SUBEND
1690 SUB WIND(W)
1700 W=INT(RND*4)-1.5 :: CAL
L GCHAR(6,4,G1):: CALL GCHAR
(6,31,G2)
1710 N=SGN(W)*4
1720 !@P-
1730 IF G1<>99 THEN CALL SPR
ITE(#15,100+W+1.5,14,41,21-N
)ELSE CALL DELSPRITE(#15)
1740 IF G2<>99 THEN CALL SPR
ITE(#16,100+W+1.5,10,41,237-
N)ELSE CALL DELSPRITE(#16)
1750 IF G2+G1=198 THEN CALL
VCHAR(8,20,116,2):: CALL SPR
ITE(#17,100+W+1.5,10,57,149-
N)
1760 !@P+
1770 SUBEND
1780 SUB TITEL(NA1$,NA2$,JOY
)
1790 GOTO 1810 :: JOY,K,S,S1
,S2
1800 !@P-
1810 DISPLAY AT(3,6):"BURGEN
SCHLACHT":".....############
##"
1820 DISPLAY AT(7,1):"DER BU
RGHERR DES BESITZES":"RABENS
TEIN HAT DEM BURGHERRN"
1830 DISPLAY AT(9,1):"VON WA
LLBERG DEN KRIEG ER-..KLAERT
."
1840 DISPLAY AT(11,1):"DIE B
EIDEN FESTUNGSWERKE":"LIEGEN
 AUF ZWEI BENACHBARTEN":"BER
GEN."
1850 DISPLAY AT(14,1):"AUF B
EIDEN BURGEN STEHT EINE":"KA
NONE."
1860 DISPLAY AT(16,1):"SIE S
OLLEN, DURCH GENAUES":"ZIELE
N UND GENAUES BEMESSEN DES P
ULVERS, DIE GEGNERISCHEBURG
DEM ERDBODEN GLEICH":"MACHEN
."
1870 DISPLAY AT(24,20):">ENT
ER<"
1880 CALL KEY(1,K,S1):: CALL
 KEY(2,K,S2):: CALL KEY(0,K,
S):: IF S=0 AND S1=0 AND S2=
0 THEN 1880
1890 CALL CLEAR
1900 IF JOY=1 THEN DISPLAY A
T(24,1):"ALLE EINGABEN MIT J
OYSTICK!!" ELSE DISPLAY AT(2
4,1):"ALLE EINGABEN UEBER TA
STATUR"
1910 DISPLAY AT(5,1):"DAS SP
IEL WIRD VON ZWEI":"SPIELERN
 GESPIELT"
1920 DISPLAY AT(12,4):"NAME
DES 1 SPIELERS:"
1930 DISPLAY AT(18,4):"NAME
DES 2 SPIELERS:"
```

LISTINGS

```
1940 CALL KEY(1,K,S1):: CALL
 KEY(2,K,S2):: IF S1=0 AND S
2=0 THEN ACCEPT AT(14,6)VALI
DATE(UALPHA)SIZE(8):NA1$ ELS
E 1960
1950 ACCEPT AT(20,6)VALIDATE
(UALPHA)SIZE(8):NA2$
1960 NA1$="RITTER "&NA1$&" V
ON RABSTEIN"
1970 NA2$="RITTER "&NA2$&" V
ON WALLBERG"
1980 !@P+
1990 SUBEND
```

LISTINGS

Fortsetzung von Seite 9

Achtung! Alle Großbuchstaben (A,D,E,F und G) sind in den folgenden Zeilen zusammen mit der CTRL-Taste einzugeben:
Teil 1: Zeilen 1380, 1410, 1420
Teil 2: Zeilen 730, 1630.

BÖRSE

Deutsches Handbuch zum Editor/Assembler DM 98; TI-LOGO kurzanleitung (28 Seiten) DM 19,80; TI-LOGO ausführliches Buch (218 Seiten DM 59; TI-Forth Diskette + Handbuch DM 148; TI-Forth Extended Basic Loader Disc DM 25; Diskettenkatalog mit Autostart DM 15; Preissenkungen für Software (alter/neuer Preis) iD Data (119) DM 75;iD Konto (119) DM 75; iD Profi Set (Konto + Data) (198) DM 140; Grahpic Master (99) DM 65; 3 D World (99) DM 65; Neue Software: Macro Word Plus Textverarbeitung DM 145; Benötigt Extended Basic, Speichererweiterung, Diskettenlaufwerk und Drucker, Macroman (Spiel) DM 40; Disassembler DM 75; Alle früheren Angebote sind hiermit ungültig. Lieferung ausschließlich per Nachnahme. Versandkosten Inland: bis DM 100 Bestellwert: 5, darüber: frei; Versandkosten Ausland: jeweilige tatsächliche Postgebühren. Hinweis: iD Programme: Bestellungen bis 1. 3. 86. Auslieferung kurzfristig. Spätere Bestellungen Auslieferung ab Juli 86 w/ Betriebsurlaub des Herstellers.

Verkaufe Hardware-Baustein zum Einbau in HF-Modul, liefert störungsfreien Video Audio-Direktausgang, dadurch besseres Bild und Ton, Info gegen Rückporto von W-Durst c/o Poeße, Gitschinerstr. 65, 1000 Berlin 61

Zu verkaufen! Module (rog. TI): Pers. Report Generator 30,–; Datenver. & Analyse 30,– The Attack 20,–; Kassette (unben): Tips & Tricks und TI 99/4A Intern je 20,– DM. J. Münzer, Tel. 0721/ 38 55 54

Verkaufe:
Haushaltsplanung Orig. TI 45,– DM, Buchungsjournal Orig. TI 75,– DM, Microsurgeon Orig. TI 40,– DM, Mind Challenger Orig. TI 20,– DM, Mini Memory, dtsch. Handbücher 250,– DM, Music maker Orig. TI 45,– DM, Datenverw. + Analyse Orig. TI 65,– DM, Text- und Dateiverw. Orig. TI 65,– DM, Statistik Orig. TI 45,– DM, Terminal Emulator II, Orig. TI 75,– DM, TI Invaders Orig. TI 30,– DM, Hustle Orig. TI 20,– DM R. Thielmann, Tel. 02234/ 79 646

Suche sehr, sehr günstig: TI 99/4A + Ext. Basic + Recorderkabel + Fernsehkabel!! Tel. 02541/46 16

Verkaufe: 99/4-Anlage, Konsole, P-Box, RS232, 32K-Ram, FD-Contr. 2 x Laufw. (doppels.) Speech, TI-Recorder + Kabel, viele Module, Software, Literatur, Matrix-Drucker (V24), Preis Verhandlungssache Tel. 06051/68 700

***** Achtung *********
Verkaufe original Extended Basic mit dt. Handbuch DM 180,–. Tel. 08806/7372

```
10 REM*********************
11 REM*      FRAGGLES      *
12 REM*                    *
13 REM*   Copyright by     *
14 REM*                    *
15 REM* H. Frielinghaus    *
16 REM*                    *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                    *
23 REM* Speicherbelegung *
24 REM*    8454 Bytes      *
25 REM*********************
26 REM
100 DATA "....MONTAG","..DIE
NSTAG","..MITTWOCH","DONNERS
TAG","...FREITAG","...SAMSTA
G","...SONNTAG"
110 RANDOMIZE
120 OPTION BASE 1
130 DIM T$(7),SX(5),SY(5)
140 CALL CLEAR
150 READ T$(1),T$(2),T$(3),T
$(4),T$(5),T$(6),T$(7)
160 CALL CHAR(96,"FFFFFFFFFF
FFFFFF")
170 CALL CHAR(97,"55AAFFFFFF
FFFFFF")
180 CALL CHAR(98,"00C0E0F0F0
F8FCFF")
190 CALL CHAR(99,"010307070F
1F3FFF")
200 CALL CHAR(100,"FFEE4404"
)
210 CALL CHAR(101,"FF7C78383
8301010")
220 CALL CHAR(102,"FF7E7C7C3
C383838")
230 CALL CHAR(103,"30101010"
)
240 CALL CHAR(104,"3C3C18244
44241C2")
250 CALL CHAR(105,"385C7C183
C3C3C3C")
260 CALL CHAR(106,"3C3C18242
2428243")
270 CALL CHAR(107,"1C3A3E183
C3C3C3C")
280 CALL CHAR(108,"1B3F3F2B1
8")
290 CALL CHAR(109,"F0F8F8F82
84848D8")
300 CALL CHAR(110,"0F1F1F1F1
412121B")
310 CALL CHAR(111,"D8FCFCD41
8")
320 CALL CHAR(112,"030F7F565
7575526")
330 CALL CHAR(113,"FFFF7E3C7
EE7C3FF")
340 CALL CHAR(114,"C0F0FE6AE
```

```
AEAAA64")
350 CALL CHAR(115,"070301")
360 CALL CHAR(116,"00FFFF")
370 CALL CHAR(117,"E0C080")
380 CALL CHAR(118,"3F7F7F777
7773F1D")
390 CALL CHAR(119,"FEFFFF777
777FEDC")
400 CALL CHAR(120,"10101010101
0101010")
410 CALL CHAR(121,"927C38FE3
8FE5438")
420 CALL CHAR(122,"1C3E6B7F4
15D7F55")
430 CALL CHAR(128,"001915151
9151519")
440 CALL CHAR(129,"009754549
7545457")
450 CALL CHAR(130,"007040406
0404040")
460 FOR I=1 TO 8
470 CALL COLOR(I,11,1)
480 NEXT I
490 CALL COLOR(9,14,1)
500 CALL COLOR(10,13,1)
510 CALL COLOR(11,15,1)
520 CALL COLOR(12,10,1)
530 CALL COLOR(13,2,16)
540 CALL SCREEN(2)
550 CALL HCHAR(5,1,96,64)
560 CALL HCHAR(13,1,96,96)
570 FOR I=5 TO 21 STEP 16
580 IF I>20 THEN 620
590 CALL HCHAR(2,I-3,72)
600 CALL HCHAR(2,I-2,73)
610 CALL HCHAR(2,I-1,45)
620 CALL HCHAR(2,I,83)
630 CALL HCHAR(2,I+1,67)
640 CALL HCHAR(2,I+2,79)
650 CALL HCHAR(2,I+3,82)
660 CALL HCHAR(2,I+4,69)
670 CALL HCHAR(2,I+5,58)
680 CALL HCHAR(2,I+6,48,5)
690 NEXT I
700 A$="DIE FRAGGLES"
710 FOR I=1 TO 12
720 CALL HCHAR(17,I,ASC(SEG$
(A$,I,1)))
730 NEXT I
740 A$="DRUECKEN SIE: <1>SPI
EL..<2>ENDE"
750 FOR I=1 TO 31
760 CALL HCHAR(23,I,ASC(SEG$
(A$,I,1)))
770 NEXT I
780 DATA 262,2,262,2,262,2,2
62,1,220,3,-6,1,-6,1,247,2,2
47,2,247,3,220,1,196,2,330,3
,294,1,262,2,0,1
790 DATA 262,2,262,2,262,2,2
62,1,220,3,-6,1,-6,1,247,2,2
47,2,220,3,247,1,262,1,0,2
800 RESTORE 780
810 READ I,J
820 IF I=0 THEN 860
830 CALL SOUND(J*200,I,6)
840 CALL KEY(0,N,M)
850 IF (N<49)+(N>50)THEN 810
 ELSE 880
860 CALL SOUND(J*200,110,30)
870 IF J=2 THEN 800 ELSE 840
880 IF N=49 THEN 900
890 END
900 CALL HCHAR(7,1,32,192)
910 CALL HCHAR(23,1,32,32)
920 CALL HCHAR(2,27,48,5)
930 P=0
940 CALL HCHAR(19,1,84)
950 CALL HCHAR(19,2,65)
960 CALL HCHAR(19,3,71)
970 CALL HCHAR(19,4,58)
980 CALL HCHAR(21,1,66)
990 CALL HCHAR(21,2,82)
1000 CALL HCHAR(21,3,73)
1010 CALL HCHAR(21,4,69,3)
1020 CALL HCHAR(21,5,70)
1030 CALL HCHAR(21,7,58)
1040 BR=0
1050 CALL HCHAR(21,15,48,3)
1060 TAG=0
1070 SG=.3
1080 REM START
1090 TAG=TAG+1+(TAG=7)*7
1100 SG=SG+SGN(.9-SG)/10
1110 FOR I=1 TO 10
1120 CALL HCHAR(19,7+I,ASC(S
EG$(T$(TAG),I,1)))
1130 NEXT I
1140 FOR BS=1 TO 4
1150 CALL HCHAR(7,1,32,192)
1160 CALL HCHAR(5,1,96,64)
1170 CALL HCHAR(13,1,97,32)
1180 X=32
1190 CALL HCHAR(11,32,105)
1200 CALL HCHAR(12,32,104)
1210 ON BS GOSUB 2690,2860,2
970,3130
1220 U=97
1230 CALL KEY(1,N,M)
1240 IF (M=0)+(N>5)THEN 1700
1250 ON N+1 GOTO 1260,1700,1
350,1700,1700,1390
1260 CALL GCHAR(12,X,S)
1270 CALL GCHAR(12,X-1,T)
1280 Q=1
1290 IF (S>111)+(T>111)+((T<
>32)*(T<104))THEN 2450
1300 IF M=-1 THEN 1780
1310 CALL HCHAR(11,X,32)
1320 CALL HCHAR(12,X,109)
1330 CALL HCHAR(12,X-1,108)
1340 GOTO 1780
```

LISTINGS

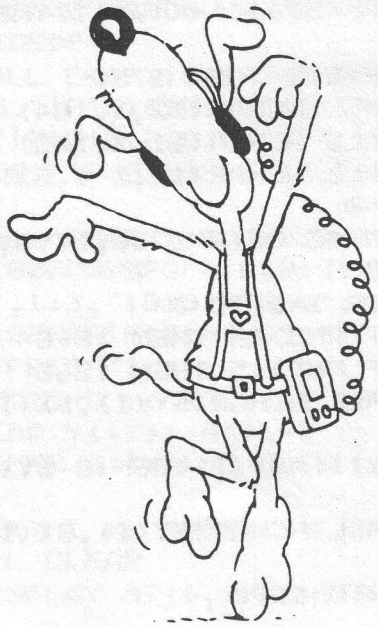
LISTINGS

```
1350 CALL VCHAR(11,X,32,2)
1360 X=X-1
1370 P=P+1
1380 GOTO 1630
1390 CALL VCHAR(11,X,32,2)
1400 X=X+SGN(1-X)
1410 CALL GCHAR(10,X,S)
1420 CALL GCHAR(11,X,T)
1430 CALL HCHAR(10,X,105)
1440 CALL HCHAR(11,X,104)
1450 IF (S<>32)+(T<>32)THEN
2450
1460 CALL VCHAR(10,X,32,2)
1470 X=X+SGN(1-X)
1480 CALL GCHAR(9,X,S)
1490 CALL GCHAR(10,X,T)
1500 CALL HCHAR(9,X,105)
1510 CALL HCHAR(10,X,104)
1520 IF (S<>32)+(T<>32)THEN
2450
1530 CALL VCHAR(9,X,32,2)
1540 X=X+SGN(1-X)
1550 CALL GCHAR(10,X,S)
1560 CALL GCHAR(11,X,T)
1570 CALL HCHAR(10,X,105)
1580 CALL HCHAR(11,X,104)
1590 IF (S<>32)+(T<>32)THEN
2450
1600 CALL VCHAR(10,X,32,2)
1610 P=P+3
1620 X=X+SGN(1-X)
1630 CALL GCHAR(13,X,U)
1640 CALL GCHAR(11,X,S)
1650 CALL GCHAR(12,X,T)
1660 CALL HCHAR(11,X,105)
1670 CALL HCHAR(12,X,104)
1680 IF (T>127)+(X<2)THEN 21
10
1690 IF (S<>32)+(T<>32)+(U<>
97)THEN 2450 ELSE 1780
1700 CALL GCHAR(11,X,S)
1710 CALL GCHAR(12,X,T)
1720 IF (S>111)+(T>111)THEN
2450
1730 IF Q=0 THEN 1780
1740 CALL HCHAR(12,X,104)
1750 CALL HCHAR(11,X,105)
1760 CALL HCHAR(12,X-1,32)
1770 Q=0
1780 ON BS GOTO 1230,1790,19
10,2020
1790 FOR I=1 TO J
1800 IF RND>SG THEN 1890
1810 IF RND>.5 THEN 1860
1820 CALL HCHAR(SY(I),SX(I),
32)
1830 SY(I)=SY(I)+SGN(8-SY(I)
)
1840 CALL HCHAR(SY(I),SX(I),
121)
1850 GOTO 1890
1860 CALL HCHAR(SY(I),SX(I),
120)
1870 SY(I)=SY(I)+SGN(11-SY(I
))
1880 CALL HCHAR(SY(I),SX(I),
121)
1890 NEXT I
1900 GOTO 1230
1910 CALL HCHAR(MY,MX,32)
1920 MX=MX+1+(MX=32)*32
1930 IF RND>SG THEN 1980
1940 IF RND>.5 THEN 1970
1950 MY=MY+SGN(11-MY)
1960 GOTO 1980
1970 MY=MY+SGN(9-MY)
1980 CALL GCHAR(MY,MX,M)
1990 IF (M>103)*(M<112)THEN
2450
2000 CALL HCHAR(MY,MX,122)
2010 GOTO 1230
2020 IF RND>SG THEN 1230
2030 CALL HCHAR(HY,19,32,2)
2040 IF RND>.5 THEN 2070
2050 HY=HY+SGN(9-HY)
2060 GOTO 2080
2070 HY=HY+SGN(12-HY)
2080 CALL HCHAR(HY,19,118)
2090 CALL HCHAR(HY,20,119)
2100 GOTO 1230
2110 P=P+BS*10
2120 NEXT BS
2130 P=P+60
2140 BR=BR+1
2150 CALL HCHAR(21,15,INT(BR
/100)+48)
2160 CALL HCHAR(21,16,INT(BR
/10)-INT(BR/100)*10+48)
2170 CALL HCHAR(21,17,BR-INT
(BR/10)*10+48)
2180 FOR I=1 TO 3
2190 FOR J=1 TO 10
2200 CALL SOUND(-500,110*J,5
)
2210 NEXT J
2220 NEXT I
2230 FOR I=HY TO 9 STEP -1
2240 IF I=HY THEN 2260
2250 CALL HCHAR(I+1,19,32,2)
2260 CALL HCHAR(I,19,118)
2270 CALL HCHAR(I,20,119)
2280 CALL SOUND(-900,110*I,0
)
2290 NEXT I
2300 CALL HCHAR(12,1,32,2)
2310 RESTORE 780
2320 FOR I=3 TO 32
2330 CALL HCHAR(11,I,107)
2340 CALL HCHAR(12,I,106)
2350 IF I>30 THEN 2410
2360 READ N,M
2370 IF N=0 THEN 2400
```

```
2380 CALL SOUND(M*200,N,5)
2390 GOTO 2410
2400 CALL SOUND(M*200,110,30
)
2410 CALL VCHAR(11,I,32,2)
2420 NEXT I
2430 CALL HCHAR(5,1,96,64)
2440 GOTO 1090
2450 CALL SOUND(2000,-6,0)
2460 FOR I=1 TO 600
2470 NEXT I
2480 A$="SPIEL-ENDE"
2490 FOR I=1 TO 10
2500 CALL HCHAR(9,I+9,ASC(SE
G$(A$,I,1)))
2510 NEXT I
2520 CALL SOUND(-600,110,25)
2530 CALL HCHAR(2,27,INT(P/1
0000)+48)
2540 CALL HCHAR(2,28,INT(P/1
000)-INT(P/10000)*10+48)
2550 CALL HCHAR(2,29,INT(P/1
00)-INT(P/1000)*10+48)
2560 CALL HCHAR(2,30,INT(P/1
0)-INT(P/100)*10+48)
2570 CALL HCHAR(2,31,P-INT(P
/10)*10+48)
2580 IF P<=RE THEN 2660
2590 RE=P
2600 CALL SOUND(-900,440,1,5
50,1,660,1)
2610 CALL HCHAR(2,11,INT(P/1
0000)+48)
2620 CALL HCHAR(2,12,INT(P/1
000)-INT(P/10000)*10+48)
2630 CALL HCHAR(2,13,INT(P/1
00)-INT(P/1000)*10+48)
2640 CALL HCHAR(2,14,INT(P/1
0)-INT(P/100)*10+48)
2650 CALL HCHAR(2,15,P-INT(P
/10)*10+48)
2660 CALL HCHAR(19,1,32,17)
2670 CALL HCHAR(21,1,32,17)
2680 GOTO 740
2690 CALL HCHAR(13,1,97,32)
2700 I=1
2710 I=I+1
2720 IF I>29 THEN 3020
2730 IF RND>SG THEN 2710
2740 IF RND>.5 THEN 2800
2750 CALL HCHAR(13,I,98)
2760 CALL HCHAR(13,I+1,32)
2770 CALL HCHAR(13,I+2,99)
2780 I=I+3
2790 GOTO 2710
2800 S=INT(RND*2)
2810 CALL HCHAR(13,I+S,96,2)
2820 CALL HCHAR(12,I+S,99)
2830 CALL HCHAR(12,I+S+1,98)
2840 I=I+3
2850 GOTO 2710
```

LISTINGS

```
2860 GOSUB 2690
2870 J=INT((SG/2+.1)*10)
2880 FOR I=1 TO J
2890 SY(I)=8
2900 SX(I)=INT(RND*32)+1
2910 CALL GCHAR(8,SX(I),S)
2920 IF S<>32 THEN 2900
2930 CALL HCHAR(7,SX(I),101)
2940 CALL HCHAR(8,SX(I),121)
2950 NEXT I
2960 GOTO 3120
2970 GOSUB 2690
2980 MY=9
2990 MX=INT(RND*10)+1
3000 CALL HCHAR(9,MX,122)
3010 GOTO 3120
3020 FOR I=1 TO 32
3030 IF RND>.5 THEN 3100
3040 IF RND>.5 THEN 3080
3050 CALL HCHAR(7,I,102)
3060 CALL HCHAR(8,I,103)
3070 GOTO 3110
3080 CALL HCHAR(7,I,101)
3090 GOTO 3110
3100 CALL HCHAR(7,I,100)
3110 NEXT I
3120 RETURN
3130 CALL HCHAR(13,1,97,32)
3140 CALL HCHAR(5,1,32,64)
3150 CALL HCHAR(7,17,112)
3160 CALL HCHAR(7,18,113)
3170 CALL HCHAR(7,19,114)
3180 CALL HCHAR(8,17,115)
3190 CALL HCHAR(8,18,116)
3200 CALL HCHAR(8,19,117)
3210 CALL HCHAR(12,1,128)
3220 CALL HCHAR(12,2,129)
3230 CALL HCHAR(12,3,130)
3240 CALL HCHAR(9,19,118)
3250 CALL HCHAR(9,20,119)
3260 HY=9
3270 RETURN
```



LISTINGS

```
10 ! ************************
11 ! *        CHORES        *
12 ! *   (Hausarbeiten)     *
13 ! *                       *
14 ! *    Copyright by       *
15 ! *                       *
16 ! *    Achim Mahnke       *
17 ! *                       *
19 ! * Benoetigte Geraete    *
20 ! *  TI99/4A  Konsole     *
21 ! *     Ext. Basic        *
22 ! *     Joystick 1        *
23 ! *                       *
26 ! *  Speicherbelegung     *
27 ! *     9172 Bytes        *
28 ! *                       *
29 ! ************************
100 CALL MAGNIFY(3):: DIM VO
R(5,5),FELD(5,5):: HISC$="00
0000" :: HISC=0 :: P=100 ::
RANDOMIZE
110 F(1),F(2)=7 :: F(3),F(4)
=5 :: F(5),F(6)=13 :: F(7),F
(8)=14 :: F(9),F(10)=4
120 CALL CHAR(96,"0000000000
0000000001021F233E0000000000
0408182444881020C000000000")
130 CALL CHAR(100,"000000000
31E2222222222233C20000000000
43CE4242424242438C000000000"
)
140 CALL CLEAR :: CALL SCREE
N(15)
150 CALL CHAR(104,"000007081
32423201F0F1616161600000000F
804E414E404F8008080800800000"
)
160 CALL CHAR(108,"000000080
4050405040504050403000000000
01010D814D414D414D810F00000"
)
170 CALL CHAR(112,"000000000
0000001020408090A0C000000000
C1C3870E0404040800000000000"
)
180 CALL CHAR(116,"000000000
0000303030300000000000000000
0000000C0C0C0C0")
190 CALL CHAR(120,"FF8080808
0808080FF0101010101010180808
080808080FF0101010101010101FF"
)
200 CALL CHAR(124,"808080808
0808080010101010101010100000
00000000FFFF"):: CALL COLOR
(3,7,1,4,7,1,12,2,1)
210 CALL CHAR(128,"00003F3F3
F3F3F3F3F3F3F3F00000000FF
CFCFCFCFCFCFCFCFCFCFCFCFC")
220 CALL CHAR(132,"000000000
0000000030000000000000000000
0000000808080E0808000000000000"
)
230 CALL CHAR(136,"00001F608
08888C4444A32221C0000000000C
040180400000000000000000000"
)
240 CALL CHAR(140,"000000001
F6080888FC5444A33221C0000000
000C020180482619149B6000000"
)
250 RESTORE 1090 :: FOR A=65
 TO 90 :: READ A$ :: CALL CH
AR(A,A$):: NEXT A
260 FOR A=48 TO 57 :: READ A
$ :: CALL CHAR(A,A$):: NEXT
A :: CALL CHAR(63,"00FE82021
E100010")
270 FOR A=3 TO 8 :: CALL COL
OR(A,5,1):: NEXT A
280 DISPLAY AT(6,11):"CHORES
" :: DISPLAY AT(20,5):"CREAT
ED BY LEMMISOFT" :: DISPLAY
AT(24,4):"PRESS ANY KEY TO B
EGIN"
290 RESTORE 1230 :: FOR A=1
TO 10 :: SU=500 :: READ Z,SP
 :: CALL SPRITE(#A,(INT(A/2+
.5)+23)*4,F(A),Z,230,0,-40)
300 CALL SOUND(-100,SU,5,SU+
5,15,SU-5,15):: SU=SU-10
310 CALL POSITION(#A,Z1,SP1)
:: IF SP1>SP+10 THEN 300
320 CALL MOTION(#A,0,0):: CA
LL LOCATE(#A,Z,SP):: NEXT A
330 FOR A=1 TO 2 :: RESTORE
1170 :: FOR B=1 TO 15 :: REA
D DA,TH :: IF DA=0 THEN GOSU
B 1600 :: GOTO 350
340 CALL SOUND(DA,TH,5,TH+5,
20,TH-5,20)
350 CALL KEY(0,K,S):: IF S<>
0 THEN 420
360 NEXT B :: NEXT A :: GOSU
B 370 :: GOTO 410
370 RESTORE 1190 :: FOR B=1
TO 24 :: READ DA,TH :: IF DA
=0 THEN GOSUB 1600 :: GOTO 3
90
380 CALL SOUND(DA,TH,5,TH+5,
20,TH-5,20)
390 CALL KEY(0,K,S):: IF S<>
0 THEN 420
400 NEXT B
410 CALL KEY(0,K,S):: IF S=0
 THEN 410
420 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL COLOR(3,7,1
,4,7,1):: FOR A=5 TO 8 :: CA
LL COLOR(A,2,1):: NEXT A
430 A$="" :: FOR A=1 TO 4 ::
 A$=A$&CHR$(127):: NEXT A ::
```

```
 B$=CHR$(120)&A$ :: A$=B$&CH
R$(121)
440 DISPLAY AT(2,12):A$ :: F
OR A=3 TO 7 :: DISPLAY AT(A,
12):"!....}" :: NEXT A :: DI
SPLAY AT(8,12):"z~~~~{"
450 FOR A=12 TO 20 STEP 2 ::
 DISPLAY AT(A,10):"xyxyxyxyx
y" :: DISPLAY AT(A+1,10):"z{
z{z{z{z{" :: NEXT A
460 SC=0 :: TR=3 :: RO=1 ::
KV=0
470 DISPLAY AT(5,2)SIZE(7):"
TRIALS:" :: DISPLAY AT(7,3)S
IZE(2):"03" :: DISPLAY AT(1,
2):"ROUND:" :: DISPLAY AT(3,
3)SIZE(2):"01"
480 DISPLAY AT(9,2)SIZE(5):"
TIME:" :: DISPLAY AT(1,21)SI
ZE(8):"HISCORE:" :: DISPLAY
AT(3,21)SIZE(6):HISC$ :: DIS
PLAY AT(5,21)SIZE(6):"SCORE:
" :: DISPLAY AT(7,21)SIZE(6)
:"000000"
490 TI=5000 :: DISPLAY AT(11
,2)SIZE(5):"5000" :: H=110
500 CALL DELSPRITE(ALL)
510 FOR A=1 TO 5 :: FOR B=1
TO 5 :: VOR(A,B)=0 :: FELD(A
,B)=0 :: NEXT B :: NEXT A
520 FOR A=1 TO 10
530 Z=INT(5*RND)+1 :: SP=INT
(5*RND)+1 :: IF VOR(Z,SP)<>0
 THEN 530 ELSE VOR(Z,SP)=A
540 CALL SPRITE(#A+8,116,F(A
),Z*10,90+SP*10):: NEXT A
550 ! ueberpruefen
560 FOR A=1 TO 5 :: FOR B=1
TO 5 :: IF VOR(A,B)<>0 THEN
D=D+1
570 IF D=5 THEN 500
580 NEXT B :: D=0 :: NEXT A
590 FOR A=1 TO 10
600 Z=INT(5*RND)+1 :: SP=INT
(5*RND)+1 :: IF FELD(Z,SP)<>
0 THEN 600 ELSE FELD(Z,SP)=A
610 CALL SPRITE(#A+18,(INT(A
/2+.5)+23)*4,F(A),73+Z*16,73
+SP*16):: NEXT A
620 FOR A=1 TO 5 :: FOR B=1
TO 5 :: IF FELD(A,B)=0 THEN
630 ELSE IF INT(FELD(A,B)/2+
.5)=INT(VOR(A,B)/2+.5)THEN K
V=KV+1
630 NEXT B :: NEXT A
640 WE=INT((RO+1)/2):: IF WE
>4 THEN WE=4
650 FOR A=1 TO WE
660 Z=INT(5*RND)+1 :: SP=INT
(5*RND)+1 :: IF FELD(Z,SP)=0
 THEN 670 ELSE 660
```

LISTINGS

```
670 FELD(Z,SP)=50 :: CALL SP
RITE(#A+1,128,2,73+Z*16,73+S
P*16):: NEXT A
680 Z=INT(5*RND)+1 :: SP=INT
(5*RND)+1 :: IF FELD(Z,SP)=5
0 THEN 680 ELSE CALL SPRITE(
#8,132,2,73+Z*16,73+SP*16)
690 GOSUB 1250 :: CALL POSIT
ION(#8,Z,SP):: CALL JOYST(1,
X,Y):: IF X=0 AND Y=0 THEN 7
30 ELSE Z=Z-Y*4 :: SP=SP+X*4
700 IF Z<89 OR Z>153 OR SP<8
9 OR SP>153 THEN Z=Z+Y*4 ::
SP=SP-X*4 :: GOTO 730 ELSE I
F GE=0 THEN CALL LOCATE(#8,Z
,SP):: GOTO 730
710 IF FELD((Z-73)/16,(SP-73
)/16)<>0 THEN Z=Z+Y*4 :: SP=
SP-X*4 :: GOTO 730 ELSE CALL
 LOCATE(#8,Z,SP):: CALL LOCA
TE(#GE,Z,SP)
720 CALL SOUND(5,500,0,-1,5)
:: CALL SOUND(5,500,0,-1,5)
730 CALL KEY(1,K,S):: IF S=0
 OR K<>18 THEN 690 ELSE CALL
 COINC(ALL,C):: CALL SOUND(5
,500,0,-1,5)
740 IF GE=0 THEN 800 ELSE ZZ
=(Z-73)/16 :: SPP=(SP-73)/16
750 IF GE<19 THEN 760 ELSE 7
70
760 GE=0 :: FELD(ZZ,SPP)=50
:: TI=TI-500 :: DISPLAY AT(1
1,1)SIZE(5):TI :: CALL SOUND
(100,700,5):: GOTO 690
770 FELD(ZZ,SPP)=GE-18 :: GE
=0
780 IF INT(FELD(ZZ,SPP)/2+.5
)=INT(VOR(ZZ,SPP)/2+.5)THEN
790 ELSE 860
790 KV=KV+1 :: IF KV=10 THEN
 900 ELSE 870
800 IF C=-1 THEN 810 ELSE 69
0
810 ZZ=(Z-73)/16 :: SPP=(SP-
73)/16 :: IF FELD(ZZ,SPP)=0
THEN 690
820 FOR A=2 TO WE+1 :: CALL
COINC(#8,#A,1,D):: IF D=-1 T
HEN GE=A :: GOTO 850
830 NEXT A :: GOTO 840
840 GE=FELD(ZZ,SPP)+18 :: IF
 INT(FELD(ZZ,SPP)/2+.5)=INT(
VOR(ZZ,SPP)/2+.5)THEN KV=KV-
1
850 FELD(ZZ,SPP)=0 :: GOTO 6
90
860 GOSUB 1430 :: GOTO 690
870 IF INT(RND*25)+1<10+RO T
HEN 880 ELSE 690
880 TI=TI-50 :: DISPLAY AT(1
```

```
1,1)SIZE(5):TI :: CALL SOUND
(100,700,5):: GOTO 690
890 ! ende der runde
900 CALL DELSPRITE(#8):: FOR
 A=1 TO WE :: CALL DELSPRITE
(#A+1):: NEXT A
910 FOR A=1 TO 10 :: CALL DE
LSPRITE(#A+8,#A+18):: SC=SC+
P :: GOSUB 1400 :: DISPLAY A
T(7,21)SIZE(6):SC$
920 CALL SOUND(50,H,0):: H=H
+10 :: FOR B=1 TO 8 :: NEXT
B :: NEXT A
930 IF TI<100 THEN 950 ELSE
TI=TI-100 :: DISPLAY AT(11,1
)SIZE(5):TI :: SC=SC+100 ::
GOSUB 1400
940 CALL SOUND(50,H,0):: H=H
+10 :: DISPLAY AT(7,21)SIZE(
6):SC$ :: GOTO 930
950 SC=SC+TI :: GOSUB 1400 :
: TI=0 :: DISPLAY AT(11,2)SI
ZE(5):"0000" :: DISPLAY AT(7
,21)SIZE(6):SC$
960 RESTORE 1190 :: FOR B=1
TO 24 :: READ DA,TH :: IF DA
=0 THEN GOSUB 1600 :: GOTO 9
80
970 CALL SOUND(DA,TH,5,TH+5,
20,TH-5,20)
980 CALL KEY(0,K,S):: IF S<>
0 THEN 1000
990 NEXT B
1000 GOSUB 1010 :: GOTO 1050
1010 DISPLAY AT(24,9):"ARE Y
OU READY?" :: FOR A=1 TO 15
:: CALL KEY(1,K,S):: IF S<>0
 AND K=18 THEN 1040
1020 NEXT A :: DISPLAY AT(24
,1):"" :: FOR A=1 TO 15 :: C
ALL KEY(1,K,S):: IF S<>0 AND
 K=18 THEN 1040
1030 NEXT A :: GOTO 1010
1040 DISPLAY AT(24,1):"" ::
FOR A=1 TO 50 :: NEXT A :: R
ETURN
1050 DISPLAY AT(24,1):"" ::
RO=RO+1 :: AN$=STR$(RO):: IF
 RO<10 THEN RO$="0"&AN$ ELSE
 RO$=AN$
1060 DISPLAY AT(3,3)SIZE(2):
RO$ :: P=P+10 :: KV=0 :: GOT
O 490
1070 END
1080 ! buchstaben
1090 DATA 00FE8282FE828282,0
0FC8282FC8282FC,00FE82808080
82FE,00FC8282828282FC,00FE80
80FE8080FE,00FE8080FE808080,
00FE80809E8282FE
1100 DATA 00828282FE828282,0
```

LISTINGS

```
038101010101038,003E22020202
82FE,00828488F0888482,008080
80808080FE,0082C6AA92828282,
0082C2A2928A8682
1110 DATA 00FE8282828282FE,0
0FE8282FE808080,00FE8282928A
86FE,00FE8282FE888482,00FE82
80FE0282FE,00FE101010101010,
00828282828282FE
1120 DATA 0082828282442810,0
0929292929292EE,008244281028
4482,0082442810101010,00FE84
08102042FE
1130 ! zahlen
1140 DATA 007E42424242427E,0
0020202020202202,007E02027E40
407E,007E02027E02027E,004242
427E020202,007E40407E02027E,
007E40407E42427E
1150 DATA 007E0202020202202,0
07E42427E42427E,007E42427E02
027E
1160 ! sound
1170 DATA 200,123,200,196,20
0,196,200,175,200,175,200,19
6,200,196,300,165,0,0
1180 DATA 200,123,200,123,15
0,155,250,147,350,165,0,0
1190 DATA 200,165,350,165,20
0,220,350,220,300,247,250,24
7,150,220,150,247,250,247,15
0,220,250,196,0,0
1200 DATA 150,196,250,165,30
0,165,150,196,250,165,300,16
5,0,0
1210 DATA 200,123,200,123,15
0,155,250,147,350,165
1220 ! spritestellen
1230 DATA 9,89,9,113,9,137,6
5,89,65,113,65,137,25,69,49,
69,25,157,49,157
1240 ! zeit
1250 TI=TI-10 :: DISPLAY AT(
11,1)SIZE(5):TI :: IF TI<=0
THEN GOSUB 1270
1260 RETURN
1270 DISPLAY AT(11,1)SIZE(5)
:" 0000" :: CALL POSITION(#8
,KA,KB):: CALL DELSPRITE(#8)
1280 FOR A=200 TO 110 STEP -
1 :: CALL SOUND(-1,A,5,A+10,
20):: NEXT A :: FOR A=6 TO 3
0 :: CALL SOUND(-1,110,A)::
NEXT A
1290 TR=TR-1 :: IF TR<0 THEN
 1310 ELSE TR$="0"&STR$(TR):
: DISPLAY AT(7,3)SIZE(2):TR$
1300 TI=5000 :: DISPLAY AT(1
1,2)SIZE(5):"5000" :: CALL S
PRITE(#8,132,2,KA,KB):: GOSU
B 1010 :: RETURN
```

LISTINGS

```
1310 SC=SC+KV*P :: GOSUB 140
0 :: DISPLAY AT(7,21)SIZE(6)
:SC$
1320 IF SC>HISC THEN 1370
1330 FOR B=1 TO 5 :: DISPLAY
 AT(24,6):"SORRY, NO HISCORE
 !" :: FOR A=1 TO 100 :: NEX
T A :: DISPLAY AT(24,1):"" :
: FOR A=1 TO 100 :: NEXT A :
: NEXT B
1340 DISPLAY AT(24,10):"NEW
GAME?" :: CALL SOUND(50,600,
5)
1350 CALL KEY(0,K,S):: IF S=
0 THEN 1350 ELSE IF K=106 TH
EN 1360 ELSE IF K=110 THEN 1
070 ELSE 1350
1360 DISPLAY AT(24,1):"" ::
GOTO 460
1370 GOSUB 1400 :: FOR B=1 T
O 8 :: DISPLAY AT(7,21)SIZE(
6):"......" :: FOR A=1 TO 40
 :: NEXT A
1380 DISPLAY AT(7,21)SIZE(6)
:SC$ :: CALL SOUND(50,900,5)
:: FOR A=1 TO 38 :: NEXT A :
: NEXT B
1390 HISC=SC :: HISC$=SC$ ::
 DISPLAY AT(3,21)SIZE(6):HIS
C$ :: FOR A=1 TO 50 :: NEXT
A :: GOTO 1340
1400 AN$=STR$(SC):: IF SC<10
00 THEN SC$="000"&AN$ ELSE I
F SC<10000 THEN SC$="00"&AN$
 ELSE IF SC<100000 THEN SC$=
"0"&AN$
1410 RETURN
1420 ! versetzen
1430 RZ=INT(5*RND)+1 :: RS=I
NT(5*RND+1):: IF FELD(RZ,RS)
=0 OR FELD(RZ,RS)=50 THEN 14
30 ELSE GG=FELD(RZ,RS)+18
1440 IF INT(FELD(RZ,RS)/2+.5
)=INT(VOR(RZ,RS)/2+.5)THEN K
V=KV-1
1450 FELD(RZ,RS)=0 :: CALL S
PRITE(#1,140,2,RZ*16+73,89):
: CALL SOUND(100,300,5)
1460 FOR A=1 TO RS-1 :: FOR
B=1 TO 50 :: NEXT B :: CALL
LOCATE(#1,RZ*16+73,(A+1)*16+
73):: CALL SOUND(100,300,5):
: NEXT A
1470 FOR A=1 TO 60 :: NEXT A
1480 CALL PATTERN(#1,136)::
FOR A=1 TO 40 :: NEXT A
1490 RZ2=INT(5*RND)+1 :: RS2
=INT(5*RND)+1 :: IF FELD(RZ2
,RS2)<>0 THEN 1490
1500 IF (RZ2=RZ)AND(RS2=RS)T
HEN 1490
1510 IF RZ2<RZ THEN MU=-16 E
LSE IF RZ<RZ2 THEN MU=16 ELS
E GOTO 1550
1520 FOR B=1 TO 10 :: NEXT B
1530 CALL POSITION(#1,PZ,PS)
:: ST=ABS(RZ2-RZ):: FOR A=1
TO ST :: CALL LOCATE(#1,PZ+M
U*A,PS):: CALL LOCATE(#GG,PZ
+MU*A,PS)
1540 CALL SOUND(100,300,5)::
 FOR B=1 TO 50 :: NEXT B ::
NEXT A
1550 IF RS2<RS THEN MU=-16 E
LSE IF RS<RS2 THEN MU=16 ELS
E GOTO 1580
1560 CALL POSITION(#1,PZ,PS)
:: ST=ABS(RS2-RS):: FOR A=1
TO ST :: CALL LOCATE(#1,PZ,P
S+MU*A):: CALL LOCATE(#GG,PZ
,PS+MU*A)
1570 CALL SOUND(100,300,5)::
 FOR B=1 TO 50 :: NEXT B ::
NEXT A
1580 FELD(RZ2,RS2)=GG-18 ::
IF INT(FELD(RZ2,RS2)/2+.5)=I
NT(VOR(RZ2,RS2)/2+.5)THEN KV
=KV+1
1590 CALL DELSPRITE(#1):: RE
TURN
1600 FOR W=1 TO 70 :: NEXT W
 :: RETURN
```

BÖRSE

Verkaufe folgende Module für den TI:

Connect Four – 4 gewinnt	25DM
Othello	25DM
Indoor Soccer	30DM
Munch-Man	27DM
Yahtzee-Kniffel	30DM
Tombstone City	27DM
Buchungsjournal	55DM
Datenverwaltung	55DM
Statistik	45DM
Parsec	35DM
Carwars	29DM
Alpiner	30DM
Miner 2049er (Spiel des Jahres 83)	50DM
Speechsynthesizer	100DM
Speech Editor (ermögl. Sprsyn in TIB)	25DM
Speechsynthesizer + Speech Editor	120DM
POLE POSITION	50DM
+ Porto bei Bestellungen unter	100DM

C. Jarque. T. 07156/34941 ab 17 h

Tausche Progr. Liste gegen Rückumschlag – Zero Zap – TI Adventure – Spiele * Speech * Editor * Bücher 07141-461147

Verk. Konsole Exbas Plus II, 32K-Ram mit centr. Min-Mem, Recorder + Kab, 4 Module, 8 Bücher, 10 TI-Journal, 11 TI-Revue + spez. 1 + 2 Cass.-Manager. Alles incl. Handbücher für 999 DM oder einzeln. Tel. 0231-20837. Stelzer Elverdinckweg 5, 4600 Dortmund 12

Verkaufe folgende orig. TI-Module:

1. Parsec	35,–
2. Schachmeister	65,–
3. Othello	25,–
4. TI-Invaders	25,–
5. Blackjack and Poker	20,–
6. Personal Record Keeping	auf Anfrage!
7. Mini Memory	auf Anfrage!

Sowie ein MBI-Interface TI-99/4A auf Centronics Preise VB!!!
Suche Tauschpartner für Assembler-Progamme sowie für Anwender-Software
Jürgen Müller
Ursfelderstr. 49
5014 Kerpen-Türnich
Tel. 02237/8278

LISTINGS

```
10 ! ************************
11 ! *       SAMURAI1       *
12 ! *       (Teil 1)       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *  Christof Hinsch     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    11363 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL SCREEN(2):: CALL CL
EAR
110 CALL CHAR(64,"FFFFFFFFFF
FFFFFF"):: DISPLAY AT(6,1):"
@@@ @@@ @@ @@ @ @ @@@ @@@ @
@...@ @ @@ @@ @ @ @ @ @ @ @
@@@ @@@ @ @ @ @ @ @ @ @@@ @
"
120 DISPLAY AT(9,1):"..@ @ @
 @...@ @ @ @@..@ @ @ @@@ @ @
 @...@ @@@ @ @ @ @ @ "
130 DISPLAY AT(14,2):"BY CH
HINSCH....JUNI 1985 " :: DIS
PLAY AT(20,2):"FUER ANLEITUN
G 'A' DRUECKEN"
140 DISPLAY AT(22,2):"ANDERN
FALLS 'S' FUER START " :: CA
LL CHAR(64,"FFFFFFFFFFFFFFFF
:: CALL CHAR(64,"FFFFFFFFFF
F"):: CALL CHAR(64,"FFFFFFFF
FF"):: CALL CHAR(64,"FFFFFFF
F")
150 CALL CHAR(64,"FFFFFF"):
 CALL CHAR(64,"FFFF")
160 CALL CHAR(64,"FF")
170 FOR COL=1 TO 12 :: CALL
COLOR(COL,14,2):: NEXT COL :
: CALL KEY(0,K,S):: IF K=115
 OR K=83 THEN 320 :: IF K=97
 OR K=65 THEN 190 :: GOTO 11
0
180 !
190 ! ANLEITUNG
200 !
210 CALL CLEAR :: DISPLAY AT
(6,1):"DIESES SPIEL FUEHRT S
IE " :: DISPLAY AT(8,1):"IN
DAS JAPAN DES 16. JAHR-" ::
DISPLAY AT(10,1):"HUNDERTS.S
IE SIND EIN RONIN,"
220 DISPLAY AT(12,1):"DER SI
CH GEGEN VERSCHIEDENE" :: DI
SPLAY AT(14,1):"SAMURAI DURC
HZUSETZEN HAT." :: DISPLAY A
T(16,1):"BEKAEMPFEN SIE SIE,
UM "
230 DISPLAY AT(18,1):"DEN TI
TEL DES SAMURAI ZU" :: DISPL
AY AT(20,1):"ERLANGEN." :: D
ISPLAY AT(24,1):"=>"
240 CALL KEY(0,K,S):: IF S=0
 THEN 240 :: CALL CLEAR
250 DISPLAY AT(2,2):"SIE KOE
NNEN DIE SAMURAI MIT" :: DIS
PLAY AT(4,2):"EINEM SCHLAGSP
RUNG KO " :: DISPLAY AT(6,2)
:"SCHLAGEN.UM ABER WEITERZU-
"
260 DISPLAY AT(8,2):"KOMMEN,
MUESSEN SIE IHR " :: DISPLAY
 AT(10,2):"SCHWERT BENUTZEN.
LASSEN" :: DISPLAY AT(12,2):
"SIE ES JEDOCH NICHT AUF"
270 DISPLAY AT(14,2):"EINEN
STELLUNGSKAMPF ANKOM-" :: DI
SPLAY AT(16,2):"MEN.(ZUM STE
CHEN MIT DEM " :: DISPLAY AT
(18,2):"SCHWERT,JOYST.1 NACH
 UNTEN"
280 DISPLAY AT(20,2):"ZIEHEN
).NUR SO ERLEGEN SIE "
290 DISPLAY AT(22,2):"IHN.
- VIEL GLUECK - "
300 DISPLAY AT(24,30):"=>"
310 CALL KEY(0,K,S):: IF S=0
 THEN 310 :: CALL CLEAR
320 CALL SCREEN(8):: BH=1
330 FOR COL=1 TO 12 :: CALL
COLOR(COL,2,8):: NEXT COL ::
 CALL CLEAR :: CALL CHAR(108
,"0F1D1A090F060F1F1F1E1F0C1E
333118000182040850 60F020C080
000000C0E0")
340 CALL CHAR(100,"00000F1D1
A090F061F3F2E3F1F197CEF00000
000800000000080E0FF60808080"
)
350 CALL CHAR(116,"010102814
120140F1F1E070103070E1CE0B0B
830E0C0F0F8F8F8F8F0B01838"
)
360 CALL CHAR(120,"070E0D143
F13171F3F371F171310000080804
080800080C16337FFF3FFEC0000"
)
370 CALL CHAR(136,"000000000
000000000C081BBAFD7A77F00000
0000000000000C0A17179F9FFBE"
)
380 CALL CHAR(140,"070F1F3F7
F7F7FFFFFFFF7F3F7FFF7F80F0F
8F8F0E0F0FCFEFFFFFEF8FCFEFE"
)
390 CALL CHAR(124,"3C746A243
C183C7EFBB9FD3F73FFB01800000
```

LISTINGS

```
0000204085060F0C00080000000"
):: CALL CHAR(97,"01030B0F1F
3FBFFF"):: CALL CHAR(98,"80C
0C0E0F8F8FDFF")
400 CALL CHAR(99,"FFFFFFFFFF
FFFFFF"):: CALL CHAR(104,"10
A262CC981135FD"):: CALL CHAR
(105,"089A92B998D9B1F5"):: C
ALL CHAR(106,"301596DBD98CD6
9F")
410 CALL CHAR(107,"FEFCF5FFA
E6DD3F7"):: CALL CHAR(112,"0
1030B0F1F3FBFFF"):: CALL CHA
R(113,"80C0C0E0F8F8FDFF"):: 
CALL CHAR(114,"FFFFFFFFFFFFF
FFF")
420 CALL CHAR(130,"003F7F7F7
F7F7F7F"):: CALL CHAR(131,"0
0FCFEFEFEFEFEFE"):: CALL CHA
R(129,"FF7E3C3C3C1818"):: CA
LL CHAR(134,"C0C0E1713B1E0C"
)
430 CALL CHAR(133,"0303878ED
C7830"):: CALL CHAR(135,"FF7
F3F1F0F070301")
440 CALL CHAR(132,"FFFEFCF8F
0E0C080")
450 CALL CHAR(35,"0707FFFFFF
070707"):: CALL COLOR(9,16,1
,10,5,16,11,5,8,2,4,8,1,4,8)
460 !
470 ! 1.SZENE
480 !
490 FOR I=1 TO 31 STEP 2 ::
CALL HCHAR(23,I,130):: NEXT
I
500 FOR I=2 TO 32 STEP 2 ::
CALL HCHAR(23,I,131):: NEXT
I
510 FOR I=1 TO 31 STEP 2 ::
CALL HCHAR(24,I,131):: NEXT
I
520 FOR I=2 TO 32 STEP 2 ::
CALL HCHAR(24,I,130):: NEXT
I :: CALL HCHAR(15,1,35,256)
:: DISPLAY AT(2,1):".......
....ab" :: DISPLAY AT(3,1):"
...........accb"
530 DISPLAY AT(4,1):".......
...accccb" :: DISPLAY AT(5,1
):".........acccccccb" :: DIS
PLAY AT(6,1):".........phiiii
iijq" :: DISPLAY AT(7,1):"..
.....prrrrrrrrrrq..ab"
540 DISPLAY AT(8,1):"......p
rrrrrrrrrrrrqaccb" :: DISPLA
Y AT(9,1):".....prrrrrrrrrrr
rrkhiijq" :: DISPLAY AT(10,1
):"....prrrrrrrrrrrrrrrrrrrr
q"
550 DISPLAY AT(11,1):"...prr
rrrrrrrrrrrrrrrrrrq" :: DI
SPLAY AT(12,1):"..prrrrrrrrr
rrrrrrrrrrrrrrrq"
560 CALL MAGNIFY(4):: CALL S
PRITE(#1,108,2,146,20)
570 CALL SPRITE(#2,116,13,14
6,220):: CALL SPRITE(#27,140
,16,25,110,0,1,#26,140,16,16
,6,0,1,#28,140,16,8,200,0,1)
580 GOSUB 690
590 CALL JOYST(1,X,Y):: CALL
 KEY(1,K,S):: IF K=18 THEN G
OSUB 820
600 IF X=0 AND Y=4 THEN GOSU
B 890
610 IF X=0 AND Y=-4 THEN GOS
UB 960
620 IF X=4 AND Y=0 THEN CALL
 MOTION(#1,0,5)
630 IF X=-4 AND Y=0 THEN CAL
L MOTION(#1,0,-1)
640 CALL COINC(#1,#2,10,L)::
 IF L=-1 THEN GOSUB 1080
650 GOTO 590 :: END
660 !
670 ! ANFANGSSOUND
680 !
690 LG=500 :: LV=250 :: LA=1
25 :: C=131 :: D=147 :: E=16
5 :: F=175 :: A=220 :: H=247
 :: C1=262
700 FOR I=1 TO 2 :: CALL SOU
ND(LV,A,0):: CALL SOUND(LV,A
,0):: CALL SOUND(LG,H,0):: N
EXT I :: CALL SOUND(LV,A,0):
: CALL SOUND(LV,H,0):: CALL
SOUND(LV,C1,0):: CALL SOUND(
LV,H,0)
710 CALL SOUND(LV,A,0):: CAL
L SOUND(LA,H,0):: CALL SOUND
(LA,A,0):: CALL SOUND(LG,F,0
):: CALL SOUND(LV,E,0)
720 CALL SOUND(LV,D,0):: CAL
L SOUND(LV,E,0):: CALL SOUND
(LV,F,0):: CALL SOUND(LV,E,0
)
730 CALL SOUND(LA,E,0):: CAL
L SOUND(LA,D,0):: CALL SOUND
(LG,C,0):: CALL SOUND(LV,A,0
):: CALL SOUND(LV,H,0)
740 CALL SOUND(LV,C1,0):: CA
LL SOUND(LV,H,0):: CALL SOUN
D(LV,A,0):: CALL SOUND(LA,H,
0):: CALL SOUND(LA,A,0):: CA
LL SOUND(LG,F,0)
750 CALL SOUND(LV,E,0):: CAL
L SOUND(LV,D,0):: CALL SOUND
(LV,E,0):: CALL SOUND(LV,F,0
):: CALL SOUND(LV,E,0):: CAL
L SOUND(LA,E,0):: CALL SOUND
(LA,D,0):: CALL SOUND(LG,C,0
```

```
)
760 FOR I=1 TO 2 :: CALL SOU
ND(LV,A,0):: CALL SOUND(LV,A
,0):: CALL SOUND(LG,H,0):: N
EXT I
770 CALL SOUND(250,30000,0):
: CALL SOUND(LG,E,0):: CALL
SOUND(LV,F,0):: CALL SOUND(L
A,H,0):: CALL SOUND(LA,A,0):
: CALL SOUND(LV,F,0):: CALL
SOUND(LG,E,0)
780 CALL MOTION(#2,0,-5):: R
ETURN :: END
790 !
800 ! SCHLAGSPRUNG
810 !
820 CALL POSITION(#1,K,L)::
CALL SPRITE(#3,120,2,K,L,0,2
5):: CALL DELSPRITE(#1):: CA
LL SOUND(-100,-4,0)
830 FOR I=1 TO 15 :: NEXT I
:: CALL COINC(#2,#3,20,0)::
IF 0=-1 THEN GOSUB 1160 :: I
F NOT 0 THEN 32767
840 CALL POSITION(#3,D,E)::
CALL SPRITE(#1,108,2,D,E)::
CALL DELSPRITE(#3):: CALL SO
UND(-100,-4,0):: RETURN :: E
ND
850 !
860 ! SPRUNG
870 !
880 CALL SOUND(-100,-4,0)
890 CALL PATTERN(#1,124):: C
ALL MOTION(#1,-10,10)
900 FOR I=1 TO 80 :: NEXT I
:: CALL MOTION(#1,9,9)
910 FOR I=1 TO 100 :: NEXT I
 :: CALL POSITION(#1,M,B)::
CALL SPRITE(#1,108,2,146,B,0
,0):: CALL SOUND(-100,-4,0):
: RETURN :: END
920 !
930 ! SCHWERTSTOSS
940 !
950 CALL SOUND(-100,-5,0)
960 CALL POSITION(#1,M,I)::
CALL SPRITE(#5,100,2,M,I)::
CALL DELSPRITE(#1)
970 CALL COINC(#5,#2,11,Q)::
 IF Q=-1 THEN GOSUB 1020 IF
NOT Q THEN 980
980 FOR I=1 TO 50 :: NEXT I
:: CALL POSITION(#5,D,Z):: C
ALL SPRITE(#1,108,2,D,Z):: C
ALL DELSPRITE(#5):: RETURN :
: END
990 !
1000 ! FEIND GESCHLAGEN
1010 !
1020 CALL MOTION(#2,0,0):: C
ALL SOUND(-300,-5,0)
1030 FOR I=1 TO 100 :: NEXT
I :: CALL PATTERN(#2,136)
1040 FOR I=1 TO 80 :: NEXT I
 :: ON BH GOSUB 1180,1190,12
00,1210,1220,1230,1820 :: CA
LL POSITION(#5,L,Z):: CALL S
PRITE(#5,108,2,L,Z):: RETURN
 :: END
1050 !
1060 ! SAMURAI GESCHLAGEN
1070 !
1080 CALL MOTION(#1,0,0,#2,0
,0):: CALL SOUND(-300,-5,0)
1090 FOR I=1 TO 100 :: NEXT
I :: CALL PATTERN(#1,136)
1100 FOR I=1 TO 400 :: NEXT
I :: CALL DELSPRITE(#1)
1110 DISPLAY AT(20,3):" EIN
NEUER START ? [J\N]" :: GOSU
B 1970
1120 CALL MOTION(#2,0,-10)
1130 CALL KEY(0,K,S):: IF K=
74 OR K=106 THEN CALL DELSPR
ITE(ALL):: CALL CLEAR :: GOT
O 320
1140 IF K=78 OR K=110 THEN C
ALL DELSPRITE(ALL):: CALL CL
EAR :: END
1150 GOTO 1130 :: END
1160 CALL SOUND(-300,-7,0)::
 CALL PATTERN(#2,136,#3,136)
:: CALL MOTION(#2,0,-7,#3,0,
7)
1170 FOR I=1 TO 100 :: NEXT
I :: CALL PATTERN(#2,116,#3,
108):: RETURN :: END
1180 CALL SPRITE(#2,116,2,14
6,256,0,-10):: BH=2 :: RETUR
N :: END
1190 CALL SPRITE(#2,116,7,14
6,256,0,-15):: BH=3 :: RETUR
N :: END
1200 CALL SPRITE(#2,116,5,14
6,256,0,-20):: BH=4 :: RETUR
N :: END
1210 CALL SPRITE(#2,116,13,1
46,256,0,-25):: BH=5 :: RETU
RN :: END
1220 CALL SPRITE(#2,116,2,14
6,256,0,-40):: BH=6 :: RETUR
N :: END
1230 CALL CLEAR :: CALL DELS
PRITE(ALL):: CALL SCREEN(8):
: DISPLAY AT(12,10):"..BRAVO
.."
1240 GOSUB 1820
1250 CALL CLEAR :: DISPLAY A
T(5,2):"SIE HABEN ALLE SAMUR
AI " :: DISPLAY AT(7,2):"ERL
EDIGT!BEWEISEN SIE NUN" :: D
```

```
ISPLAY AT(9,2):"IM ENTSCHEID
UNGSKAMPF IHR"
1260 DISPLAY AT(11,2):"KOENN
EN UND SIE WERDEN DEM" :: DI
SPLAY AT(13,2):"DAIMYO DER P
ROVINZ SAGAMI" :: DISPLAY AT
(15,2):"VORGEFUEHRT!"
1270 DISPLAY AT(22,8):"..AUF
 GEHT'S.."
1280 CALL KEY(0,K,S):: IF S=
0 THEN 1280
1290 !
1300 ! 2.SZENE
1310 !
1320 CALL SCREEN(5)
1330 CALL CHAR(41,"00000018"
)
1340 CALL CHAR(48,"03070F1E3
E3E7E7E")
1350 CALL CHAR(96,"030285422
3110D0F0703070F1F0F0F00C0E07
060C080C0E0F0D8E8F87CF4C440"
)
1360 CALL CHAR(51,"7E7E3E3E1
E0F0703")
1370 FOR COL=1 TO 12 :: CALL
 COLOR(COL,2,5):: NEXT COL :
: CALL CLEAR :: CALL COLOR(1
1,2,5,8,2,5,13,2,5,2,6,5,14,
12,5,3,16,5,1,6,5)
1380 DISPLAY AT(3,1):"...0 )
...)...FE...)......))......
.....) rr" :: DISPLAY AT(4,1
):"...3........prrq..).." ::
 DISPLAY AT(5,1):"..)...)...
.prrrrq" :: DISPLAY AT(6,1):
"......)...prrrrrrq"
1390 DISPLAY AT(7,1):"..)...
.) prrrrrrrrq )..) )" :: DIS
PLAY AT(8,1):"..)..)..prrrrr
rrrrrq" :: DISPLAY AT(9,1):"
 )....prrrrrrrrrrrrq..)" ::
 DISPLAY AT(10,1):"..)...prr
rrrrrrrrrrrrq..)"
1400 DISPLAY AT(11,1):"..)..
prrrrrrrrrrrrrrrrq..)..) )"
:: DISPLAY AT(12,1):" )..prr
rrrrrrrrrrrrrrrrq...)" :: DI
SPLAY AT(13,1):")..prrrrrrrr
rrrrrrrrrrrrq"
1410 DISPLAY AT(14,1):") prr
rrrrrrrrrrrrrrrrrrrrrq )" ::
DISPLAY AT(15,1):" prrDAAAAA
AAGrDAAAAAAAAGrrq"
1420 DISPLAY AT(16,1):"E..r.
.......r.........r..F" ::
DISPLAY AT(17,1):"   r#######
###r##########r" :: DISPLAY
AT(18,1):"...r##########r####
######r"
1430 DISPLAY AT(19,1):"...r#
########r##########r" :: DIS
PLAY AT(20,1):"...r#########
r##########r" :: DISPLAY AT(
21,1):"...r#########r#######
###r"
1440 DISPLAY AT(22,1):"...r#
########r##########r"
1450 FOR I=1 TO 31 STEP 2 ::
 CALL HCHAR(23,I,130):: NEXT
 I
1460 FOR I=2 TO 32 STEP 2 ::
 CALL HCHAR(23,I,131):: NEXT
 I
1470 FOR I=1 TO 31 STEP 2 ::
 CALL HCHAR(24,I,131):: NEXT
 I
1480 FOR I=2 TO 32 STEP 2 ::
 CALL HCHAR(24,I,130):: NEXT
 I
1490 CALL SPRITE(#1,108,2,14
6,47,#2,116,2,146,180,0,-12)
1500 CALL JOYST(1,X,Y):: CAL
L KEY(1,K,S)
1510 IF K=18 AND X=-4 AND Y=
0 THEN GOSUB 1690
1520 IF K=18 THEN GOSUB 820
1530 IF X=0 AND Y=-4 THEN GO
SUB 1610
1540 IF X=4 AND Y=0 THEN CAL
L MOTION(#1,0,5)
1550 IF X=0 AND Y=4 THEN GOS
UB 890
1560 IF X=-4 AND Y=0 THEN CA
LL MOTION(#1,0,-2)
1570 CALL COINC(#1,#2,10,K):
: IF K=-1 THEN GOSUB 1080
1580 CALL COINC(#1,#27,10,U)
:: IF U=-1 THEN GOSUB 1080
1590 GOTO 1500
1600 END
1610 CALL SOUND(-100,-5,0)::
 CALL POSITION(#1,M,V):: CAL
L SPRITE(#5,100,2,M,V):: CAL
L DELSPRITE(#1)
1620 CALL PATTERN(#2,96):: C
ALL MOTION(#2,-25,0)
1630 FOR I=1 TO 50 :: NEXT I
 :: CALL MOTION(#2,25,0)
1640 FOR I=1 TO 50 :: NEXT I
 :: CALL POSITION(#2,0,S)::
CALL SPRITE(#2,116,2,146,S,0
,-10)
1650 CALL DELSPRITE(#6):: CA
LL POSITION(#5,G,A):: CALL S
PRITE(#1,108,2,G,A):: CALL D
ELSPRITE(#5)
1660 CALL COINC(#1,#27,10,L)
:: IF L=-1 THEN GOSUB 1060
1670 RETURN
1680 END
1690 CALL MOTION(#1,0,-20)
```

```
1700 FOR I=1 TO 150 :: NEXT
I
1710 CALL COINC(#1,#2,16,E)
: IF E=-1 THEN GOSUB 1740
1720 IF E=1 THEN CALL MOTION
(#1,0,5)
1730 RETURN :: END
1740 CALL SOUND(-300,-5,0)::
 CALL PATTERN(#2,136):: CALL
 MOTION(#2,0,0):: CALL MOTIO
N(#1,0,5)
1750 FOR I=1 TO 200 :: NEXT
I
1760 CALL PATTERN(#2,116)::
CALL MOTION(#2,0,-15)
1770 BH=7 :: GOTO 590
1780 END
1790 !
1800 ! TUSCH
1810 !
1820 CALL SOUND(2000,30000,2
0):: CALL SOUND(200,131,0)::
 CALL SOUND(200,165,0):: CAL
L SOUND(200,196,0):: CALL SO
UND(800,262,0)
1830 CALL DELSPRITE(#2)
1840 CALL SOUND(100,262,0)::
 CALL SOUND(100,196,0):: CAL
L SOUND(100,262,0):: CALL SO
UND(100,196,0):: CALL SOUND(
100,165,0)
1850 CALL SOUND(100,165,0)::
 CALL SOUND(100,175,0):: CAL
L SOUND(100,175,0):: CALL SO
UND(20,20000,30):: CALL SOUN
D(1000,165,6,196,0,262,0)
1860 IF BH=7 THEN 1890
1870 RETURN
1880 END
1890 CALL KEY(0,K,S):: IF S=
0 THEN 1890
1900 FOR COL=1 TO 14 :: CALL
 COLOR(COL,12,2):: NEXT COL
1910 CALL SCREEN(2)
1920 CALL CLEAR :: CALL DELS
PRITE(ALL)
1930 DISPLAY AT(17,1)BEEP:"*
 LADEN SIE NUN BITTE DAS..":
 :"..ZWEITE PROGRAMM EIN !"
1940 CALL KEY(0,K,S):: IF S=
0 THEN 1940
1950 RUN "CS1"
1960 END
1970 FOR I=1 TO 2 :: FOR U=0
 TO 8 :: CALL SOUND(-100,165
,U):: NEXT U :: NEXT I
1980 CALL SOUND(10,30000,0):
: FOR I=1 TO 2 :: FOR U=0 TO
 8 :: CALL SOUND(-100,147,U)
:: NEXT U :: NEXT I
1990 CALL SOUND(10,30000,0):
: FOR I=1 TO 2 :: FOR U=0 TO
 8 :: CALL SOUND(-100,131,U)
:: NEXT U :: NEXT I :: CALL
SOUND(10,30000,0)
2000 FOR U=0 TO 8 :: CALL SO
UND(-100,147,U):: NEXT U
2010 RETURN :: END
```

LISTINGS

BÖRSE

HI-Res-Grafik+Sprites für TI99/4A-Minimem. 27 neue BASIC-Bef (PLOT, SETTO, LINE, SPRITE-Bef. etc.). Keine Einschr. d. TI-BASIC! DM 30,– M. Kausch, Semmelweisstr. 151, 5 Köln 80

Verk. TI 99/4A, Peribox m. Diskl. + Contr. 32 K Erweit., Schnittst. RS232; Drucker Seikos. GP550A + Kabel; Recorder + Ka., Joyst. div. Spiele + Disk., Buchjourn., Finber., Datenverw. VHB 3200 DM, Tel. 04563/8900

Wer verschenkt C-64 oder anderen COMMODORE??? Das Ding darf uralt und total kaputt sein!!! Schicken Sie das Gerät bitte an Hubert Mühlbacher, Plattenberg 3, 8221 Waging (Unkosten werden erstattet)

Verkaufe: Original-TI-Messestand + 16 Module. Tel.: 0202/72 862

Verkauf wegen Systemwechsel Konsole 200,– ; X-Bas II + 200,–; 32K-RAM extern mit centr. 200,–; Min-Mem 180,–; Recorder Sony + Kabel 180,–; 99 Spezial I und II 50,–; Module Statistik, Al. ADD, Min-Mis., Othello 50,–; Q-Save Plat aus Chip 25,–; TI-Revue 10 Hefte 20,–; TI-Journal 10 Hefte 20,–; TI-Spezial I u. II 10,–; Versch. Bücher 20,–; Programmsamml. 1 + 2 Journal 20,–; Handbuch incl. Schaltplan für Konsole 10,–; Kass-Kassetten-Manager 20,–; alles zusammen nur 999,– DM. S. Stelzer, Elverdinckweg 5, 4600 Dortmund 12. Tel. 0231-20837 ab 17 h

---

Wer schickt TI-Anfänger kostenlose Listings o. Spiele. Zuschriften an R. Dupre, Hühnerstraße 4, 4270 Dorsten 1.

++++ R. G. SOFTGAME ++ SPITZENSOFTWARE IN TI OD. EX INFO FÜR 1.50 DM BEI R. GOERKE ++++ WATZLING 10, 8250 DORFEN ++++++++++++++++++++++

SUCHE LITERATUR ETC ÜBER SPEECHSYNT. TEL. 08081/3236

Verk. TI99/4A+Ext. Basic m. Handb. + Recorder + Kabel + Joyst. + Joyst. Adapter + 4 Module (Parsec, Tomb. City, Munch: M., Indoor S. + 1 Buch + ca. 100 Prg. u. S TI Revues für DM 450,– ab 14 h; Tel.: 0203/772247, Duisburg 1.

Buchungsjournal (Modul) 79,–; Datenverw. + Anal. (Modul) 59,–; Lagerverwaltung (Disk) 59,–; Rechnungsstellung (Disk) 59,–; Statistik (Modul) 35,–; Vokabeltrainer (Cass.) 10,–; TI Basic Kurs (engl. Cass.) 10,–. R. Unruh, Telefon 0711/719267.

WER TAUSCHT 32-K-ERWEITERUNG GEGEN 110 SUPERPROGRAMME z. B. SLICKS, TENNIS, SUPERFROG ET C. INTERESSENTEN BITTE MELDEN BEI SCHNEIDER MANFRED TEL. 06874/6705.

Verkaufe super TI-Progr.! In TI- und EX-Basic für nur DM 1,50. Interessenten melden bei Schneider Manfred, Nunkircher Straße 42a, 6648 Wadernbüschfeld/Saar. Oder Telefon 06874/6705.

Tausche 110 Superprogramme gegen Disclaufwerk 6106. Meldet Euch bei: Schneider Manfred, Telefon 06874/6705.

Ich bekomme von euch RS 232-Schnittstelle und ihr bekommt von mir 110 Superprogramme. Meldet euch bei Schneider, Manfred, Telefon 06874/6705.

LISTINGS

```
10 ! ************************
11 ! *       SAMURAI2       *
12 ! *       (Teil 2)       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *  Christoph Hinsch    *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    10737 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CLEAR
110 FOR COL=1 TO 14 :: CALL
COLOR(COL,12,2):: NEXT COL
120 CALL SCREEN(2)
130 DISPLAY AT(3,8):"GUTE AR
BEIT !"
140 DISPLAY AT(6,2):"SIE SIN
D EINER DER WENIGEN,"
150 DISPLAY AT(8,2):"DER DIE
 EHRE HAT,DEM DAIMYO"
160 DISPLAY AT(10,2):"SEIN K
OENNEN ZU BEWEISEN."
170 DISPLAY AT(12,2):"SOLLTE
N SIE DIESE PRUEFUNG "
180 DISPLAY AT(14,2):"BESTEH
EN,SO WERDEN SIE ALS"
190 DISPLAY AT(16,2):"SAMURA
I IN DIE DIENSTE DES"
200 DISPLAY AT(18,2):"SHOGUN
S AUFGENOMMEN."
210 DISPLAY AT(21,2):"BEGEBE
N SIE SICH NUN AUF"
220 DISPLAY AT(23,2):"DEN WE
G ZUM PALAST !"
230 CALL KEY(0,K,S):: IF S=0
 THEN 230
240 CALL CLEAR
250 DISPLAY AT(4,2):"OH,BEIN
AHE HAETT ICH'S VER-"
260 DISPLAY AT(6,2):"GESSEN.
"
270 DISPLAY AT(10,2):"SEIEN
SIE AUF DER HUT !"
280 DISPLAY AT(12,2):"SIE HA
BEN NUN NICHT MEHR "
290 DISPLAY AT(14,2):"GEGEN
'HARMLOSE' SAMURAI ZU"
300 DISPLAY AT(16,2):"KAEMPF
EN !"
310 DISPLAY AT(18,2):"KYUDO-
KAEMPFER SIND IHRE"
320 DISPLAY AT(20,2):"GEGNER
(JAP.BOGENSCHUETZEN)."
330 DISPLAY AT(22,2):"NEHMEN
SIE DEM VOGEL DEN"
340 DISPLAY AT(24,2):"SCHLUE
SSEL ZUR BURG AB !"
350 CALL KEY(0,K,S):: IF S=0
 THEN 350
360 CALL CLEAR :: DISPLAY AT
(12,9)BEEP:"BITTE WARTEN !"
370 !
380 CALL CHAR(108,"030706060
30103070D3F0F0103070603C040A
040C080C0E0F0FFF8C0E030180C"
)
390 CALL CHAR(136,"000000000
000000000C081BBAFD7A77F00000
0000000000000C0A17179F9FFBE"
)
400 CALL CHAR(124,"3C746A243
C183C7EFBB9FD3F73FFB01800000
0000204085060F0C00080000000"
)
410 CALL CHAR(116,"01010A112
12020313FFF202123272F18E070B
830E0C0E0F0F8F8E8F8FCFEFF03"
)
420 CALL CHAR(140,"000000000
00004081F0804000000000000000
0000000002CF82C")
430 CALL CHAR(100,"000000000
81D3F6F3F0F0300000000000183
C78F0E0E6FEFCE6C0C060000000"
)
440 CALL CHAR(120,"070E0D143
F13171F3F371F17131000008080 4
080800080C16337FFF3FFEC0000"
)
450 CALL CHAR(97,"01030B0F1F
3FBFFF"):: CALL CHAR(98,"80C
0C0E0F8F8FDFF"):: CALL CHAR(
99,"FFFFFFFFFFFFFFFF")
460 CALL CHAR(104,"10A262CC9
81135FD"):: CALL CHAR(105,"0
89A92B998D9B1F5"):: CALL CHA
R(106,"301596DBD98CD69F")
470 CALL CHAR(107,"FEFCF5FFA
E6DD3F7"):: CALL CHAR(112,"0
1030B0F1F3FBFFF"):: CALL CHA
R(113,"80C0C0E0F8F8FDFF")
480 CALL CHAR(129,"0103070F1
F070301"):: CALL CHAR(114,"F
FFFFFFFFFFFFFF"):: CALL CHA
R(130,"003F7F7F7F7F7F7F")
490 CALL CHAR(42,"080808FF08
0808FF"):: CALL CHAR(131,"00
FCFEFEFEFEFEFE"):: CALL CHAR
(135,"FF7E3C3C3C1818"):: CAL
L CHAR(134,"C0C0E1713B1E0C")
500 CALL CHAR(133,"0303878ED
C7830"):: CALL CHAR(135,"FF7
F3F1F0F070301"):: CALL CHAR(
132,"FFFEFCF8F0E0C080"):: CA
LL CLEAR
```

LISTINGS

```
510 CALL SCREEN(8)
520 FOR COL=1 TO 14 :: CALL
COLOR(COL,2,8):: NEXT COL
530 CALL COLOR(9,16,1,10,2,1
6,11,2,8,2,13,8)
540 FOR I=1 TO 31 STEP 2 ::
CALL HCHAR(23,I,130):: NEXT
I
550 FOR I=2 TO 32 STEP 2 ::
CALL HCHAR(23,I,131):: NEXT
I
560 FOR I=1 TO 31 STEP 2 ::
CALL HCHAR(24,I,131):: NEXT
I
570 FOR I=2 TO 32 STEP 2 ::
CALL HCHAR(24,I,130):: NEXT
I
580 CALL VCHAR(1,31,114,48)
590 DISPLAY AT(2,1):".......
ab.................p"
600 DISPLAY AT(3,1):"......a
ccb...............pr"
610 DISPLAY AT(4,1):".....ac
cccb.............prr"
620 DISPLAY AT(5,1):"....phi
iiijq...........prrr"
630 DISPLAY AT(6,1):"...prrr
rrrrrq.........prrrr"
640 DISPLAY AT(7,1):"..prrrr
rrrrrrq.......prrrrr"
650 DISPLAY AT(8,1):" prrrrr
rrrrrrrq.....prrrrrr"
660 DISPLAY AT(9,1):"prrrrrr
rrrrrrrrq....prrrrrrr"
670 DISPLAY AT(10,1):".....
.............prrrrrrrr"
680 DISPLAY AT(11,1):"******
**********prrrrrrrrr"
690 DISPLAY AT(12,1):"******
**********prrrrrrrrrr"
700 DISPLAY AT(13,1):"******
**********prrrrrrrrrrr"
710 DISPLAY AT(14,1):"******
*********prrrrrrrrrrrr"
720 FOR I=1 TO 2 :: CALL VCH
AR(11,I,42,5):: NEXT I
730 DISPLAY AT(15,1):"******
**********GrD******G"
740 FOR I=16 TO 22 :: DISPLA
Y AT(I,1):"................
..r" :: NEXT I
750 CALL KEY(0,K,S):: IF S=0
 THEN 750
760 CALL MAGNIFY(4):: CALL S
PRITE(#2,116,2,146,200,#1,10
8,2,146,20)
770 CALL SPRITE(#10,140,2,14
6,190,0,-30)
780 CALL SPRITE(#15,100,2,12
0,250,0,-25)
790 CALL JOYST(1,X,Y):: CALL
KEY(1,K,S)
800 IF K=18 THEN GOSUB 920
810 IF X=0 AND Y=-4 THEN GOS
UB 1100
820 IF X=0 AND Y=4 THEN GOSU
B 1010
830 IF X=4 AND Y=0 THEN CALL
 MOTION(#1,0,10)
840 IF X=-4 AND Y=0 THEN CAL
L MOTION(#1,0,-5)
850 CALL COINC(#10,146,5,10,
P):: IF P=-1 THEN CALL DELSP
RITE(#10):: CALL SPRITE(#10,
140,2,146,190,0,-30)
860 CALL COINC(#1,#2,10,R)::
 IF R=-1 THEN GOTO 1190
870 CALL COINC(#1,#10,10,L):
: IF L=-1 THEN GOSUB 1170
880 CALL COINC(#1,146,220,10
,L):: IF L=-1 AND FR=9 THEN
GOTO 1360
890 GOTO 790
900 END
910 !
920 ! SCHLAGSPRUNG
930 !
940 CALL POSITION(#1,K,L)::
CALL SPRITE(#3,120,2,K,L,0,2
5):: CALL DELSPRITE(#1):: CA
LL SOUND(-100,-4,0)
950 FOR I=1 TO 15 :: NEXT I
:: CALL COINC(#2,#3,20,O)::
IF O=-1 THEN GOSUB 970 :: IF
 NOT O THEN 960
960 CALL POSITION(#3,D,E)::
CALL SPRITE(#1,108,2,D,E)::
CALL DELSPRITE(#3):: CALL SO
UND(-100,-4,0):: RETURN :: E
ND
970 CALL SOUND(-300,-7,0)::
CALL PATTERN(#2,136,#3,136):
: CALL MOTION(#3,0,10)
980 CALL DELSPRITE(#10):: TT
=11
990 FOR I=1 TO 200 :: NEXT I
 :: CALL PATTERN(#3,108):: R
ETURN :: END
1000 !
1010 ! SPRUNG
1020 !
1030 CALL SOUND(-100,-4,0)::
 CALL PATTERN(#1,124):: CALL
 MOTION(#1,-10,10)
1040 FOR I=1 TO 100 :: NEXT
I :: CALL MOTION(#1,9,9)
1050 CALL COINC(ALL,O):: IF
O=-1 THEN CALL POSITION(#1,H
,G):: CALL SPRITE(#1,136,2,1
46,G):: GOTO 1190
1060 FOR I=1 TO 100 :: NEXT
I :: CALL POSITION(#1,M,B)::
```

```
 CALL SPRITE(#1,108,2,146,B,
0,0):: CALL SOUND(-100,-4,0)
:: RETURN :: END
1070 !
1080 ! VOGELFLUG
1090 !
1100 CALL POSITION(#15,A,Q):
: CALL SPRITE(#16,100,2,A,Q,
7,-7):: CALL DELSPRITE(#15,#
10)
1110 FOR I=1 TO 150 :: NEXT
I :: CALL COINC(#1,#16,5,U):
: IF U=-1 THEN CALL SOUND(-2
00,990,0):: FR=9
1120 CALL MOTION(#16,-7,-7)
1130 FOR I=1 TO 100 :: NEXT
I
1140 IF TT<>11 THEN CALL SPR
ITE(#10,140,2,146,190,0,-30)
1150 CALL POSITION(#16,K,F):
: CALL SPRITE(#15,100,2,120,
F,0,-25):: CALL DELSPRITE(#1
6):: RETURN :: END
1160 !
1170 ! SAMURAI GESCHLAGEN
1180 !
1190 CALL MOTION(#1,0,0,#2,0
,0):: CALL SOUND(-300,-5,0)
1200 CALL DELSPRITE(#10)
1210 FOR I=1 TO 100 :: NEXT
I :: CALL PATTERN(#1,136)
1220 FOR I=1 TO 200 :: NEXT
I
1230 CALL SOUND(-1000,220,0,
221,9,222,10)
1240 CALL MOTION(#2,0,-10)
1250 DISPLAY AT(20,4):"EIN N
EUER START?[J/N]"
1260 CALL KEY(0,K,S)
1270 IF K=78 OR K=110 THEN C
ALL DELSPRITE(ALL):: CALL CL
EAR :: END
1280 IF K=74 OR K=106 THEN 1
300
1290 GOTO 1260
1300 CALL CLEAR :: CALL DELS
PRITE(ALL):: CALL CHARSET ::
 CALL SCREEN(2):: FOR COL=1
TO 14 :: CALL COLOR(COL,12,2
):: NEXT COL
1310 DISPLAY AT(16,1)BEEP:"*
 SPIELE CASSETTE AUF ": :"..
PROGRAMM 1 ZURUECK,": :"..DA
NN DRUECKE ENTER"
1320 CALL KEY(0,K,S):: IF S=
0 THEN 1320
1330 ON BREAK NEXT :: RUN "C
S1"
1340 CALL KEY(0,K,S):: IF S=
0 THEN 1340
1350 END
```

LISTINGS

```
1360 CALL CLEAR :: CALL DELS
PRITE(ALL)
1370 CALL CHAR(42,"FFFFFFFFF
FFFFFFF",115,"787878787878787
78")
1380 CALL CHAR(140,"0302C562
33190F0301070F030F3F3F1FC0E0
7060C080C0E0F0D0F0C0E0F8FCF4
")
1390 CALL CHAR(100,"00000F1D
1A090F061F3F2E3F1F197CEF0000
0000800000000080E0FF60808080
")
1400 CALL CHAR(120,"00000100
0000000010103060C071F1F0FF0B8
5C98F060E0F0F0F8E8F8E0FCFEFA
")
1410 CALL CHAR(136,"0F1D1A09
0F060F1F1F1E1F0C1E3331180001
8204085060F020C080000000C0E0
")
1420 FOR COL=1 TO 14 :: CALL
 COLOR(COL,12,2):: NEXT COL
1430 CALL SCREEN(2)
1440 DISPLAY AT(2,1):"- SIE
HABEN ES GESCHAFFT -"
1450 DISPLAY AT(5,1):"SIE SI
ND IN DEN PALAST VOR-"
1460 DISPLAY AT(7,1):"GEDRUN
GEN."
1470 DISPLAY AT(10,1):"SIE W
ERDEN NUN DEM DAIMYO"
1480 DISPLAY AT(12,1):"GEGEN
UEBERGESTELLT."
1490 DISPLAY AT(14,1):"ICH H
OFFE SIE BENEHMEN SICH"
1500 DISPLAY AT(16,1):"GUT,.
.-..SAMURAI..-"
1510 DISPLAY AT(18,1):"ES WE
RDEN IHNEN DIE ZWEI "
1520 DISPLAY AT(20,1):"SCHWE
RTER UEBERREICHT."
1530 DISPLAY AT(23,1):"....-
..GRATULIERE..-"
1540 CALL KEY(0,K,S):: IF S=
0 THEN 1540
1550 CALL CLEAR :: CALL SCRE
EN(12)
1560 FOR COL=1 TO 14 :: CALL
 COLOR(COL,7,12):: NEXT COL
:: CALL COLOR(2,8,8)
1570 CALL VCHAR(1,31,114,48)
:: CALL VCHAR(1,1,114,48)
1580 CALL HCHAR(1,1,114,64)
1590 FOR I=1 TO 31 STEP 2 ::
 CALL HCHAR(23,I,130):: NEXT
 I
1600 FOR I=2 TO 32 STEP 2 ::
 CALL HCHAR(23,I,131):: NEXT
 I
1610 FOR I=1 TO 31 STEP 2 ::
```

```
 CALL HCHAR(24,I,131):: NEXT
 I
1620 FOR I=2 TO 32 STEP 2 ::
 CALL HCHAR(24,I,130):: NEXT
 I
1630 DISPLAY AT(3,1):"D.....
.....G"
1640 DISPLAY AT(7,1):".....
.....p"
1650 DISPLAY AT(8,1):"...pq.
..pq.....pq.....pr"
1660 DISPLAY AT(9,1):"..prrq
 prrq....prrq.....prr"
1670 DISPLAY AT(10,1):"..s**
s s**s....s**s.....prrr"
1680 DISPLAY AT(11,1):"..s**
s s**s....s**s....prrrr"
1690 DISPLAY AT(12,1):"..s**
s s**s....s**s...prrrrr"
1700 DISPLAY AT(13,1):"..s**
s s**s....s**s..prrrrrr"
1710 FOR I=14 TO 16 :: DISPL
AY AT(I,1):"..s**s s**s....s
**s" :: NEXT I
1720 DISPLAY AT(17,1):"..rrr
r rrrr....rrrr"
1730 DISPLAY AT(20,1):".....
.....rrrrrrr"
1740 DISPLAY AT(21,1):".....
.....rrrrrrrr"
1750 DISPLAY AT(22,1):".....
.....rrrrrrrrr"
1760 CALL MAGNIFY(4)
1770 CALL SPRITE(#2,120,2,12
2,200,#1,108,2,146,20)
1780 CALL KEY(1,K,S):: CALL
JOYST(1,X,Y)
1790 IF X=4 AND Y=0 THEN CAL
L MOTION(#1,0,5)
1800 IF X=-4 AND Y=0 THEN CA
LL MOTION(#1,0,-5)
1810 IF X=0 AND Y=-4 THEN GO
SUB 1840
1820 CALL COINC(#1,146,134,1
,T):: IF T=-1 THEN GOTO 1900
1830 GOTO 1780
1840 CALL POSITION(#1,D,A)::
 CALL SPRITE(#3,100,2,D,A)::
 CALL DELSPRITE(#1)
1850 FOR I=1 TO 300 :: NEXT
I
1860 CALL POSITION(#3,P,O)::
 CALL SPRITE(#1,108,2,P,O)::
 CALL DELSPRITE(#3):: RETURN
 :: END
1870 END
1880 CALL KEY(0,K,S):: IF S=
0 THEN 1880
1890 END
1900 CALL MOTION(#1,0,0):: C
ALL PATTERN(#1,100):: CALL P
```

LISTINGS

```
ATTERN(#2,140)
1910 FOR I=1 TO 100 :: NEXT
I
1920 FOR I=1 TO 15 :: CALL P
ATTERN(#2,120)
1930 FOR G=1 TO 100 :: NEXT
G :: CALL PATTERN(#2,140)::
CALL SOUND(-10,-5,0)
1940 FOR G=1 TO 100 :: NEXT
G :: NEXT I
1950 GOSUB 2050
1960 CALL MOTION(#1,0,-9)
1970 CALL COINC(#1,146,10,1,
0):: IF O=-1 THEN CALL PATTE
RN(#1,108):: CALL DELSPRITE(
ALL):: CALL CLEAR :: GOTO 19
90
1980 GOTO 1970
1990 CALL CHARSET :: FOR COL
=1 TO 14 :: CALL COLOR(COL,1
2,2):: NEXT COL :: CALL SCRE
EN(2):: DISPLAY AT(8,1):"SIE
 HABEN ES GESCHAFFT."
2000 DISPLAY AT(11,1):"SOEBE
N WURDEN SIE ZUM"
2010 DISPLAY AT(13,1):"SAMUR
AI DES DAIMYOS ERNANNT."
2020 DISPLAY AT(20,9):"- END
E -"
2030 CALL KEY(0,K,S):: IF S=
0 THEN 2030
2040 CALL CLEAR :: END
2050 D=147 :: E=165 :: G=196
 :: A=220 :: H=247 :: C1=262
 :: D1=294 :: E1=330 :: G1=3
92
2060 LGP=750 :: LG=500 :: LV
=250 :: LA=125 :: LS=65
2070 FOR I=1 TO 2
2080 CALL SOUND(375,D,0):: C
ALL SOUND(LA,E,0):: CALL SOU
ND(LS,D,0):: CALL SOUND(375,
E,0):: CALL SOUND(375,G,0)
2090 CALL SOUND(LA,A,0):: CA
LL SOUND(LS,G,0):: CALL SOUN
D(375,A,0):: CALL SOUND(LV,H
,0):: CALL SOUND(LA,A,0):: C
ALL SOUND(LA,H,0)
2100 CALL SOUND(LA,A,0):: CA
LL SOUND(LA,G,0):: CALL SOUN
D(LA,E,0):: CALL SOUND(375,D
,0):: CALL SOUND(LA,E,0):: C
ALL SOUND(LA,D,0)
2110 CALL SOUND(LA,E,0):: CA
LL SOUND(375,G,0):: CALL SOU
ND(LA,A,0):: CALL SOUND(LA,G
,0):: CALL SOUND(LA,A,0):: C
ALL SOUND(LA,H,0)
2120 CALL SOUND(LA,A,0):: CA
LL SOUND(LA,G,0):: CALL SOUN
D(LA,E,0):: CALL SOUND(375,G
```

```
,0):: CALL SOUND(LA,A,0):: C
ALL SOUND(LV,G,0)
2130 CALL SOUND(LGP,G,0):: C
ALL SOUND(LG,D1,0):: CALL SO
UND(LV,D1,0):: CALL SOUND(LV
,D1,0,G1,0):: CALL SOUND(LV,
H,0,E1,0)
2140 CALL SOUND(LV,A,0,D1,0)
2150 CALL SOUND(LA,A,0):: CA
LL SOUND(LA,H,0):: CALL SOUN
D(LA,A,0):: CALL SOUND(LA,G,
0)
2160 CALL SOUND(LA,E,0):: CA
LL SOUND(375,D,0):: CALL SOU
ND(LA,E,0):: CALL SOUND(LA,D
,0):: CALL SOUND(LA,E,0)
2170 CALL SOUND(375,G,0):: C
ALL SOUND(LA,A,0):: CALL SOU
ND(LA,G,0):: CALL SOUND(LA,A
,0)
2180 CALL SOUND(LA,H,0):: CA
LL SOUND(LA,A,0):: CALL SOUN
D(LA,G,0):: CALL SOUND(LA,E,
0):: CALL SOUND(LA,D,0):: CA
LL SOUND(LA,E,0)
2190 CALL SOUND(375,G,0):: C
ALL SOUND(LA,A,0):: CALL SOU
ND(LV,G,0):: CALL SOUND(LGP,
G,0)
2200 CALL SOUND(1000,30000,1
0)
2210 NEXT I :: RETURN :: END
```

BÖRSE

Verkaufe TI 99/4A zum Ausschlachten. Handbuch, Trafo, Modulator, Konsole und Tastatur! Dirk Kronshage, Akazienstraße 24, 4092 Bad Salzuflen oder Telefon 05222/13837.

Verkaufe: 1. Personal Record Keeping DM 50; 2. Personal Report Generator DM 50; 3. Individual Accounting (Buchungsjournal) DM 70; 4. Meteor Belt (Sprachausg.) DM 40; 5. Burger Time DM 30; 6. Moon Patrol DM 30; suche Modulexpander !!!. Telefon 06181/23399 (ab 20 h). B. Pompe, 6450 Hanau 1, Postfach 2041.

Verkaufe TI99/4A + P-Box + 32 K + Controller + Dsklw. + RS-232 + Ex.-Basic + Joystick + Schachmod. + Apesoft-Grafic + Lit.: Ed./Ass., 99Special II. für DM 1.900. Telefon 02361/26460.

Gebe Masch. Progr. (ED/ASS) auf Kassette ab. Nur EX + 32 K nötig. — Infos gegen Rückporto. — B. Dusny, Mülldorfer Straße 31, 5205 St. Augustin 1.

Suche günstig Drucker Epson RX-80 oder FX-80. Telefon 02241/330700 (abends).

Editor/Assembler auf Kassette: Nur DM 60,– Exb. + 32 k notwendig. Lothar Krauß, Tel. 0561/88 71 29

Verk. Miniassembler + E/A Handbuch + viele Progm. + viele Infos für DM 120,– Tel. 06134/45 19

EINMALIG! 128 KB für 10 DM. Eine MC randvoll mit TI 99/4A Programmen in TI- und ExBasic über 128 KB ** keine Kopien DC Mondweg 9 5450 Neuwied 1

DIE VERFLIXTE BÖRSE

Die verflixte Börse ist ein nicht ganz ernst zu nehmendes Börsenspiel.
Sinn des Spieles, einer Mischung aus Börsenspiel und Monopoly, ist es, mit einem Startkapital von 2000 DM durch An- und Verkauf zum Millionär zu werden. Zur Auswahl stehen 8 AG's mit einem Mindestkurs von 50,– DM und einem Höchstkurs von 999,– DM.
Im Anzeigenfeld werden der Name der AG, der momentane Kurs, die Anzahl der Aktienbesitze und der letzte Kurs, zu dem Aktien gekauft wurden, angezeigt. Letzteres ist für den Kursvergleich nicht ohne Interesse.
Die Hürden, die man während des Spieles nehmen muß, können positiv oder negativ sein. Am gefährlichsten ist der „Börsenkrach". Hier können die Aktienkurse für einen Durchgang unter die 50,– DM-Grenze fallen. Doch die eben noch so günstig gekauften Aktien können einer durch den Börsenkrach pleitegegangenen AG gehören, und die Aktienbesitze sind verloren.
Das Spiel besteht aus 3 Teilen: Kaufen, Ereignis, Verkaufen. Will man nicht kaufen oder verkaufen, so muß man eine "0" eingeben. Will man kaufen oder verkaufen, dann muß man zuerst die Nummer der AG, dann die Anzahl der Aktien eingeben.
Nach jedem "Kaufen" werden das Barkapital mit 4% verzinst und die Aktienkurse neu festgelegt.
Um jedoch nicht in Versuchung zu geraten, nur durch Verzinsung das Ziel zu erreichen, wird man nach 4 Durchgängen, in denen weder gekauft noch verkauft wurde oder ein Ereignis eingetreten ist, mit einem Teilbarkapitalverlust bestraft.
Ebenso wird man bestraft, wenn man Aktien verkaufen will, die man nicht besitzt.
Hat man Schulden und ist kein Kapital vorhanden, so ist man gezwungen, die Schulden durch Aktienverkauf zu tilgen. Ist das nicht möglich, so ist das Spiel verloren.
Hat man ein Barkapital von 1 000 000,– DM erreicht, so werden alle Aktien verkauft und der Endstand errechnet.

Variablenliste:

V<>	=	Aktienbesitz
W<>	=	Aktienkurs
N$<>	=	AG – Name
PR<>	=	Aktienkaufkurs
HK	=	zu erreichendes Limit
KP<>	=	Kapital-Plus-Rechnung
KM<>	=	Kapital-Minus-Rechnung
RN<>	=	Zufallszahl
K	=	Barkapital
ZA	=	An- und Verkaufskontrolle
A	=	Hilfsvariable
@,@1	=	Hilfsvariable
BO	=	Kontrolle für Börsensturz
AK,B	=	Hilfsvariable

Erklärung der Sub-Unterprogramme:

SUB KU<W<>>
Anzeige der Aktienkurse
SUB PR<PR<>>
Anzeige des Kaufpreises der Aktien
SUB BE<V<>>
Anzeige der Aktienbesitze
SUB KO<K<>>
Anzeige des Barkapitals
SUB CL
Teillöschung des Bildschirmes
SUB P <@>
Zählschleife als Pause
SUB FE<K,@>
Fehlerbehandlung (Bestrafung)
SUB TI
Titelbild
SUB TA
Aufforderung zur Tasteneingabe

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *    DIE VERFLIXTE     *
13 ! *       BOERSE         *
14 ! *                      *
15 ! *    Copyright by      *
16 ! *                      *
17 ! *    Peter  Rueck      *
18 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     9850 Bytes       *
28 ! *                      *
29 ! ************************
100 DIM V(8),W(8),N$(8),PR(8
):: HK=999999 :: RANDOMIZE :
: ON WARNING NEXT :: CALL CH
AR(124,RPT$("08",8),128,"000
0FFFF")
110 DEF KP(X)=K+INT(K/100*X)
:: DEF KM(X)=K-INT(K/100*X):
: DEF RN(X)=INT(RND*X)+1
120 DATA BP,SHELL,AEG,AUDI,T
ALBOT,IBM,D.BANK,D.BENZ
130 FOR I=1 TO 8 :: READ N$(
I):: NEXT I :: CALL CLEAR ::
 CALL SCREEN(2):: FOR I=0 TO
 14 :: CALL COLOR(I,16,2)::
NEXT I
140 IMAGE "######"
150 IMAGE "###"
160 IMAGE "#####"
170 IMAGE "####### DM"
180 CALL TI :: FOR I=1 TO 14
 :: CALL COLOR(I,2,16):: NEX
T I :: CALL COLOR(0,14,16,3,
14,16,4,14,16)
190 K=2000 :: FOR I=1 TO 8 :
: PR(I),W(I)=MAX(50,RN(150))
:: V(I)=0 :: NEXT I
200 DISPLAY AT(1,11)ERASE AL
L:"KURS ! BES. !EK-PR" :: CA
LL HCHAR(2,3,128,28):: CALL
VCHAR(3,18,124,9):: CALL VCH
AR(3,25,124,9):: CALL HCHAR(
12,3,128,28)
210 FOR I=1 TO 8 :: DISPLAY
AT(3+I,1)SIZE(9):STR$(I)&".
"&N$(I):: NEXT I :: CALL KU(
W()):: CALL BE(V()):: CALL P
R(PR()):: DISPLAY AT(14,1):"
KONTO :"
220 CALL KO(K):: ZA=ZA+1 ::
IF RND<.1 THEN 600 ELSE IF B
O=1 THEN 830
230 IF K<0 THEN 1230
240 IF ZA>4 THEN 860
250 CALL CL :: DISPLAY AT(17
,1):"WAS WOLLEN SIE KAUFEN ?
" :: ACCEPT AT(17,28)SIZE(1)
VALIDATE("012345678")BEEP:A
260 IF A=0 THEN 280 ELSE DIS
PLAY AT(19,1):"WIEVIEL ?" ::
 ACCEPT AT(19,11)VALIDATE(DI
GIT)BEEP:B :: IF K-W(A)*B<0
OR W(A)=0 THEN CALL FE(K,0):
: GOTO 250
270 ZA=0 :: V(A)=V(A)+B :: K
=K-W(A)*B :: PR(A)=W(A):: CA
LL BE(V()):: CALL PR(PR()):
 CALL KO(K):: GOTO 230
280 CALL CL :: FOR I=1 TO 8
:: W(I)=W(I)+INT(RND*51-RND*
51):: W(I)=MAX(W(I),50):: W(
I)=MIN(W(I),999):: NEXT I ::
 CALL KU(W())
290 K=KP(4):: CALL KO(K):: F
OR I=1 TO 8 :: IF V(I)>0 THE
N 310
300 NEXT I :: GOTO 250
310 IF RND>.6 THEN 390
320 IF RND<.4 THEN 880
330 CALL CL :: IF K<0 THEN 1
230
340 DISPLAY AT(17,1):"WAS WO
LLEN SIE VERKAUFEN ?" :: ACC
EPT AT(17,28)SIZE(1)VALIDATE
("012345678")BEEP:A :: IF A=
0 THEN 220
350 IF V(A)=0 THEN CALL FE(K
,0):: GOTO 330
360 ZA=0 :: DISPLAY AT(19,1)
:"WIEVIEL ?" :: ACCEPT AT(19
,11)VALIDATE(DIGIT)BEEP:B ::
 IF V(A)<B THEN CALL FE(K,0)
:: GOTO 330
370 K=K+W(A)*B :: V(A)=V(A)-
B :: CALL BE(V()):: CALL KO(
K):: IF K>HK THEN 1480
380 GOTO 330
390 ON RN(22)GOTO 400,430,45
0,480,490,510,520,540,560,57
0,600,620,630,650,660,690,71
0,720,740,750,770,790
400 IF V(1)<1 THEN 330
410 DISPLAY AT(17,1)BEEP:"BP
-AKTIEN SIND ZUSAMMENGE-..BR
OCHEN. AUCH IHRE SIND DA- MI
T VERLOREN."
420 V(1),W(1),PR(1)=0 :: CAL
L KU(W()):: CALL BE(V()):: C
ALL PR(PR()):: GOTO 820
430 @=RN(30):: DISPLAY AT(17
,1)BEEP:"SIE HABEN EINE ERBS
CHAFT VON":STR$(@)&" SHELL A
KTIEN GEMACHT.": :"GRATULIER
E !!!"
440 V(2)=V(2)+@ :: PR(2)=W(2
):: CALL PR(PR()):: CALL BE(
```

LISTINGS

```
V()):: GOTO 820
450 IF V(3)<2 THEN 330
460 DISPLAY AT(17,1)BEEP:"SI
E MUESSEN DIE HAELFTE":"IHRE
R AEG AKTIEN ZUR TILGUNGALTE
R SCHULDEN VERWENDEN."
470 V(3)=V(3)-INT(V(3)/2)::
CALL BE(V()):: GOTO 820
480 @=MAX(1000,RN(5000)):: D
ISPLAY AT(17,1)BEEP:"GRATULI
ERE !!!": :"SIE HABEN "&STR$
(@)&" DM IM LOTTO": :"GEWONN
EN." :: K=K+@ :: CALL KO(K):
: GOTO 820
490 IF W(4)>899 THEN 330
500 DISPLAY AT(17,1)BEEP:"IH
RE AUDI AKTIEN SIND UM": :"1
00 DM GESTIEGEN." :: W(4)=W(
4)+100 :: CALL KU(W()):: GOT
O 820
510 @=RN(45):: DISPLAY AT(17
,1)BEEP:"DURCH INFLATION VER
RINGERT": :"SICH IHR KAPITAL
 UM "&STR$(@)&" %" :: K=KM(@
):: CALL KO(K):: GOTO 820
520 IF V(5)<1 OR W(5)<150 TH
EN 330
530 DISPLAY AT(17,0)BEEP:"TA
LBOT AKTIEN SIND UM 100 DM":
 :"GEFALLEN !!!" :: W(5)=W(5
)-100 :: CALL KU(W()):: GOTO
 820
540 IF V(6)<1 OR W(6)<100 TH
EN 330
550 DISPLAY AT(17,1)BEEP:"IB
M AKTIEN SIND UM 50 %": :"GE
FALLEN !!!" :: W(6)=W(6)-INT
(W(6)/100*50):: CALL KU(W())
:: GOTO 820
560 @=MAX(500,RN(2000)):: DI
SPLAY AT(17,1)BEEP:"SIE ERHA
LTEN EINE DIVIDENDE": :"VON
"&STR$(@)&" DM" :: K=K+@ ::
CALL KO(K):: GOTO 820
570 IF V(8)<1 THEN 330
580 DISPLAY AT(17,1)BEEP:"SI
E SIND GEZWUNGEN, IHRE":"DAI
MLER AKTIEN ZUM JETZIGEN KUR
S ZU VERKAUFEN."
590 K=K+V(8)*W(8):: V(8)=0 :
: CALL BE(V()):: CALL KO(K):
: GOTO 820
600 CALL CL :: @=RN(100):: D
ISPLAY AT(17,1)BEEP:"BOERSEN
STURZ !!!": :"ALLE AKTIEN SI
ND UM "&STR$(@)&" %": :"GEFA
LLEN !!!"
610 FOR I=1 TO 8 :: W(I)=W(I
)-INT(W(I)/100*@):: W(I)=MAX
(W(I),1):: NEXT I :: CALL KU
(W()):: BO=1 :: GOTO 820
620 DISPLAY AT(17,1)BEEP:"BE
IM POKERN VERLIEREN SIE": :"
5000 DM !!!" :: K=K-5000 ::
CALL KO(K):: GOTO 820
630 IF V(4)=1 OR W(4)<250 TH
EN 330
640 DISPLAY AT(17,1)BEEP:"AU
DI AKTIEN SIND UM 200 DM": :
"GEFALLEN !!!" :: W(4)=W(4)-
200 :: CALL KU(W()):: GOTO 8
20
650 DISPLAY AT(17,1)BEEP:"BE
IM PFERDERENNEN GEWINNEN SIE
": :"2500 DM !!!" :: K=K
+2500 :: CALL KO(K):: GOTO 8
20
660 IF V(5)<1 THEN 330
670 DISPLAY AT(17,1)BEEP:"UM
 VERLUSTE ZU VERMEIDEN....MU
ESSEN SIE IHRE TALBOT.....AK
TIEN ABSTOSSEN"
680 K=K+V(5)*W(5):: V(5)=0 :
: CALL BE(V()):: CALL KO(K):
: GOTO 820
690 @=RN(25):: DISPLAY AT(17
,1)BEEP:"SIE ERHALTEN "&STR$
(@)&" DAIMLER": :"AKTIEN VER
ERBT."
700 V(8)=V(8)+@ :: PR(8)=W(8
):: CALL BE(V()):: CALL PR(P
R()):: GOTO 820
710 DISPLAY AT(17,1)BEEP:"BE
IM WETTEN VERLIEREN SIE": :"
DIE HALFTE IHRES KAPITALS !"
 :: K=KM(50):: CALL KO(K)::
GOTO 820
720 IF V(8)<1 OR W(8)>800 TH
EN 330
730 DISPLAY AT(17,1)BEEP:"DA
IMLER AKTIEN SIND UM": :"150
 DM GESTIEGEN !!!" :: W(8)=W
(8)+150 :: CALL KU(W()):: GO
TO 820
740 DISPLAY AT(17,1)BEEP:"ZU
M GEBURTSTAG ERHALTEN SIE":
:"1000 DM." :: K=K+1000 :: C
ALL KO(K):: GOTO 820
750 IF V(1)<1 OR W(8)>850 TH
EN 330
760 DISPLAY AT(17,1)BEEP:"BP
 AKTIEN SIND UM 100 DM": :"G
ESTIEGEN !!!" :: W(1)=W(1)+1
00 :: CALL KU(W()):: GOTO 82
0
770 IF V(7)<1 OR W(7)<200 TH
EN 330
780 DISPLAY AT(17,1)BEEP:"DE
UTSCHE BANK AKTIEN SIND": :"
UM 100 DM GEFALLEN !!!" :: W
(7)=W(7)-100 :: CALL KU(W())
:: GOTO 820
```

LISTINGS

```
790 DISPLAY AT(17,1)BEEP:"DA
S FINANZAMT VERLANGT 35 % IH
RES KAPITALS UND 10 %.....IH
RER AKTIEN ALS EINKOMMEN- ST
EUER."
800 K=KM(35):: FOR I=1 TO 8
:: IF V(I)<1 THEN 810 ELSE @
=INT(V(I)/100*10):: V(I)=V(I
)-MAX(@,1)
810 NEXT I :: CALL BE(V()):
 CALL KO(K):: GOTO 820
820 CALL TA :: CALL CL :: GO
TO 330
830 CALL CL :: BO=0 :: @=RN(
8):: IF V(@)<1 THEN BO=1 ::
GOTO 250
840 DISPLAY AT(17,1)BEEP:N$(
@)&" IST DURCH DEN BOERSEN":
 :"STURZ PLEITE !!!": :"DEIN
E AKTIEN SIND VERFALLEN."
850 V(@),W(@),PR(@)=0 :: CAL
L KU(W()):: CALL PR(PR()):
CALL BE(V()):: IF RND<.2 THE
N 830 ELSE 820
860 ZA=0 :: @=RN(50):: DISPL
AY AT(17,1)BEEP:"S T R A F E
..! ! !": :STR$(@)&" % IHRES
 KAPITALS WERDEN": :"ABGEZOG
EN."
870 K=KM(@):: CALL KO(K):: G
OTO 820
880 ON RN(14)GOTO 890,950,98
0,1020,1040,1120,1170,1200,1
290,1300,1310,1350,1390,1420
890 AK=0 :: DISPLAY AT(17,1)
BEEP:"SIE SIND GEZWUNGEN 25
%.....IHRER AKTIEN ZU VERKAU
FEN...UM STEUERSCHULDEN ABZU
DECKEN" :: CALL P(800)
900 FOR I=1 TO 8 :: AK=AK+V(
I):: NEXT I :: AK=INT(AK/100
*25):: AK=MAX(AK,1)
910 CALL CL :: DISPLAY AT(17
,1)BEEP:"SIE MUESSEN "&STR$(
AK)&" AKTIEN":"VERKAUFEN !!!
" :: DISPLAY AT(20,1):"WAS S
OLL VERKAUFT WERDEN ?"
920 ACCEPT AT(20,28)SIZE(1)V
ALIDATE("12345678")BEEP:A ::
 IF V(A)=0 THEN CALL FE(K,0)
:: GOTO 910
930 DISPLAY AT(22,1):"WIEVIE
L ?" :: ACCEPT AT(22,11)VALI
DATE(DIGIT)BEEP:B :: IF V(A)
<B THEN CALL FE(K,0):: GOTO
910
940 V(A)=V(A)-B :: CALL BE(V
()):: AK=AK-B :: IF AK>0 THE
N 910 ELSE GOTO 820
950 @=RN(8):: IF V(@)<1000 T
HEN 330
960 DISPLAY AT(17,1)BEEP:"SI
E SIND HAUPTAKTIONAER BEI":N
$(@)&" UND ERHALTEN SOMIT EI
NE":"DIVIDENDE VON 5 DM PRO"
:"AKTIE"
970 K=K+V(@)*5 :: CALL KO(K)
:: GOTO 820
980 @=RN(8):: IF K<(INT(W(@)
/2))*300 THEN 330
990 DISPALY AT(17,1)BEEP:N$(
@)&" AKTIEN - NOTVERTRKAUF":
 :"SIE ERHALTEN DIE CHANCE 3
00 AKTIEN ZUR HAELFTE DES":"
AKTIENKURSES ZU KAUFEN."
1000 DISPLAY AT(24,1):"WOLLE
N SIE ?...[J/N].....J" ::
ACCEPT AT(24,28)SIZE(-1)VALI
DATE("JN")BEEP:AN$ :: IF AN$
="N" THEN 330
1010 V(@)=V(@)+300 :: K=K-(I
NT(W(@)/2))*300 :: CALL BE(W
()):: CALL KO(K):: GOTO 820
1020 @=RN(150):: @1=RN(8)::
IF V(@1)<@ THEN 330
1030 DISPLAY AT(17,1)BEEP:"S
IE WURDEN UM "&STR$(@)&" "&N
$(@1):"AKTIEN BESTOHLEN !!!"
 :: V(@1)=V(@1)-@ :: CALL BE
(V()):: GOTO 820
1040 @=RN(8):: @1=25 :: DISP
LAY AT(17,1)BEEP:N$(@)&" AKT
IEN SIND AUF 25 DM":"GEFALLE
N" :: W(@)=25 :: CALL KU(W()
)
1050 DISPLAY AT(20,1):"WIEVI
EL KAUFEN ?" :: ACCEPT AT(20
,18)VALIDATE(DIGIT)BEEP:A ::
 IF A=0 THEN 1070 ELSE IF K<
A*25 THEN CALL FE(K,0):: GOT
O 1050
1060 K=K-@1*A :: PR(@)=@1 ::
 V(@)=V(@)+A :: CALL BE(V())
:: CALL PR(PR()):: CALL KO(K
):: GOTO 1100
1070 IF V(@)<1 THEN 1100 ELS
E DISPLAY AT(20,1):"WIEVIEL
VERKAUFEN ?"
1080 ACCEPT AT(20,21)VALIDAT
E(DIGIT)BEEP:A :: IF A=0 THE
N 1100 ELSE IF V(@)<A THEN C
ALL FE(K,0):: GOTO 1080
1090 V(@)=V(@)-A :: K=K+25*A
 :: CALL BE(V()):: CALL KO(K
)
1100 IF @1<26 THEN 820 ELSE
IF RND<.1 THEN 820 ELSE CALL
 CL
1110 DISPLAY AT(17,1)BEEP:N$
(@)&" IST PLEITE !!!": :"ALL
E AKTIEN SIND VERLOREN." ::
W(@),V(@),PR(@)=0 :: CALL KU
```

```
(W()):: CALL BE(V()):: CALL
PR(PR()):: GOTO 820
1120 @=RN(8):: @1=RN(8):: IF
 @=@1 THEN 1120 ELSE IF V(@)
<1 THEN 330
1130 DISPLAY AT(17,1)BEEP:"S
IE SIND GEZWUNGEN IHRE":N$(@
)&" AKTIEN ZU VERKAUFEN":"UN
D ERHALTEN DAFUER":N$(@1)&"
AKTIEN" :: A,B=0
1140 A=V(@)*W(@):: V(@)=0 ::
 IF A>W(@1)THEN 1150 ELSE B=
W(@1)-A :: K=K-B :: V(@1)=V(
@1)+1 :: PR(@1)=W(@1):: CALL
 BE(V()):: CALL PR(PR()):: C
ALL KO(K):: GOTO 820
1150 B=B+1 :: A=A-W(@1):: IF
 A>W(@1)THEN 1150
1160 V(@1)=V(@1)+B :: K=K+A
:: PR(@1)=W(@1):: CALL BE(V(
)):: CALL PR(PR()):: CALL KO
(K):: GOTO 820
1170 DISPLAY AT(17,1)BEEP:"S
IE ERHALTEN PRO AKTIE 4 %":
:"DIVIDENTE AUSBEZAHLT." ::
FOR I=1 TO 8 :: IF V(I)=0 TH
EN 1190
1180 K=K+INT(W(I)/100*4)*V(I
)
1190 NEXT I :: CALL KO(K)::
GOTO 820
1200 DISPLAY AT(17,1)BEEP:"S
IE MUESSEN PRO AKTIE 4,5 %":
 :"MAKLERGEBUEHREN ZAHLEN !!
!" :: @=4.5 :: FOR I=1 TO 30
0 :: NEXT I
1210 FOR I=1 TO 8 :: IF V(I)
=0 THEN 1220 ELSE K=K-INT(W(
I)/100*@)*V(I)
1220 NEXT I :: CALL KO(K)::
IF K>=0 THEN 820
1230 B=ABS(K):: CALL CL :: D
ISPLAY AT(17,1)BEEP:"IHR KAP
ITAL REICHT NICHT AUSES FEHL
EN "&STR$(B)&" DM":"DIE DURC
H AKTIENVERKAUF.....BEGLICHE
N WERDEN MUESSEN."
1240 DISPLAY AT(22,1):"WAS W
OLLEN SIE VERKAUFEN ?" :: AC
CEPT AT(22,28)SIZE(1)VALIDAT
E("12345678")BEEP:A :: IF V(
A)<1 THEN CALL FE(K,1):: GOT
O 1230
1250 DISPLAY AT(24,1):"WIEVI
EL ?" :: ACCEPT AT(24,11)VAL
IDATE(DIGIT)BEEP:B :: IF V(A
)<B THEN CALL FE(K,1):: GOTO
 1230
1260 K=K+W(A)*B :: V(A)=V(A)
-B :: CALL BE(V()):: CALL KO
(K):: IF K<0 THEN 1270 ELSE
820
1270 FOR I=1 TO 8 :: IF V(I)
>0 THEN 1230
1280 NEXT I :: GOTO 1450
1290 DISPLAY AT(17,1)BEEP:"S
IE MUESSEN 3,5 % VER-": :"MO
EGENSTEUER PRO AKTIE": :"ZAH
LEN !!!" :: @=3.5 :: GOTO 12
10
1300 @=RN(8):: @1=RN(100)::
DISPLAY AT(17,1)BEEP:N$(@)&"
 IST UM "&STR$(@1)&" %":"GEF
ALLEN" :: @1=INT(W(@)/100*@1
):: W(@)=@1 :: CALL KU(W()):
: GOTO 1050
1310 A=INT(W(1)/100*75)*25+I
NT(W(3)/100*75)*25+INT(W(5)/
100*75)*25+INT(W(7)/100*75)*
25 :: IF A>K THEN 330
1320 DISPLAY AT(17,1)BEEP:"S
IE KOENNEN EIN AKTIENPAKET..
VON 25 BP-,25 AEG-,25 TALBOT
UND 25 D.BANK-AKTIEN FUER...
75 % DER AKTIENKURSE KAUFEN.
PREIS = "&STR$(A)&" DM"
1330 DISPLAY AT(24,1):"WOLLE
N SIE KAUFEN ? [J/N]..J" ::
ACCEPT AT(24,28)SIZE(-1)VALI
DATE("JN")BEEP:AN$ :: IF AN$
="N" THEN 330
1340 K=K-A :: FOR I=1 TO 7 S
TEP 2 :: V(I)=V(I)+25 :: NEX
T I :: CALL BE(V()):: CALL K
O(K):: GOTO 820
1350 FOR I=1 TO 8 :: IF V(I)
>100 THEN 1370
1360 NEXT I :: GOTO 330
1370 DISPLAY AT(17,1)BEEP:"S
IE ERHALTEN PRO 100 AKTIEN E
INE SONDER-DIVIDENTE VON...2
 AKTIEN ! ! !" :: FOR I=1 TO
 8 :: IF V(I)>100 THEN V(I)=
V(I)+INT(V(I)/100)*2
1380 NEXT I :: CALL BE(V()):
: GOTO 820
1390 @=RN(8):: IF V(@)<800 T
HEN 330
1400 DISPLAY AT(17,1)BEEP:N$
(@)&" - AKTIONAERSVER-":"SAM
MLUNG":"SIE WURDEN IN DEN VO
RSTAND..GEWAEHLT UND GEBEN D
ESSHALB EIN FEST. DAS FEST K
OSTET...SIE 5500 DM."
1410 K=K-5500 :: CALL KO(K):
: GOTO 820
1420 @=RN(8):: IF V(@)<500 T
HEN 330
1430 DISPLAY AT(17,1)BEEP:N$
(@)&" IST IN SCHWIERIG-":"KE
ITEN.": :"ALS AKTIONAER UNTE
RSTUETZEN SIE DIE FIRMA MIT
```

```
15000 DM." :: K=K-15000 :: C
ALL KO(K):: GOTO 820
1440 CALL CL :: DISPLAY AT(2
2,1)BEEP:"SIE BETRUEGEN. STR
AFE 50 %": :"IHRES KAPITALS
!!!" :: K=KM(50):: CALL KO(K
):: GOTO 820
1450 CALL CL :: DISPLAY AT(1
7,1)BEEP:"SIE HABEN SICH LEI
DER VER-..SPEKULIERT UND SIN
D SOMIT AMENDE."
1460 DISPLAY AT(24,1):"NEUES
 SPIEL ? [J/N].....J" :: AC
CEPT AT(24,28)SIZE(-1)VALIDA
TE("JN")BEEP:@$
1470 IF @$="J" THEN CALL CLE
AR :: GOTO 190 ELSE CALL CLE
AR :: STOP
1480 CALL CL :: FOR I=1 TO 8
 :: K=K+W(I)*V(I):: NEXT I
1490 DISPLAY AT(17,1)BEEP:"G
ESCHAFT !!!":"SIE SIND MILLI
ONAER.":"IHR ERARBEITETES KA
PITAL BE-TRAEGT DM"&STR$(K)
1500 GOTO 1460
1510 SUB KU(W()):: FOR I=1 T
O 8 :: DISPLAY AT(3+I,12)SIZ
E(3):USING 150:W(I):: NEXT I
 :: SUBEND :: SUB PR(PR()):
 FOR I=1 TO 8 :: DISPLAY AT(
I+3,26)SIZE(3):USING 150:PR(
I)
1520 NEXT I :: SUBEND :: SUB
 BE(V()):: FOR I=1 TO 8 :: D
ISPLAY AT(3+I,17)SIZE(5):USI
NG 160:V(I):: NEXT I :: SUBE
ND
1530 SUB KO(K):: DISPLAY AT(
14,9):USING 170:K :: SUBEND
:: SUB CL :: CALL HCHAR(17,1
,32,256):: SUBEND :: SUB P(@
):: FOR I=1 TO @ :: NEXT I
1540 SUBEND :: SUB FE(K,@)::
 CALL CL :: DISPLAY AT(22,1)
BEEP:"SIE BETRUEGEN. STRAFE
50%": :"IHRES KAPITALS !!!"
:: IF @=0 THEN K=K-INT(K/100
*50)ELSE K=K+INT(K/100*50)
1550 CALL KO(K):: CALL P(800
):: CALL CL :: SUBEND :: SUB
 TI :: CALL MAGNIFY(2):: DIS
PLAY AT(2,3)BEEP:"PR-SOFT PR
AESENTIERT :"
1560 A$="DIE VERFLIXTE BOERS
E" :: A=40 :: B=0 :: FOR I=1
 TO 20 :: B$=SEG$(A$,I,1)::
IF B$=" " THEN A=A+40 :: B=0
 :: GOTO 1580
1570 CALL SPRITE(#I,ASC(B$),
16,A,40):: B=B+16 :: CALL LO
CATE(#I,A+B/4,40+B)
```

LISTINGS

```
1580 NEXT I :: DISPLAY AT(24
,4)BEEP:"C 1985 BY PETER RUE
CK" :: CALL P(800):: CALL TA
 :: CALL DELSPRITE(ALL):: CA
LL CLEAR :: SUBEND :: SUB TA
1590 DISPLAY AT(24,1):">> TA
STE DRUECKEN <<" :: CALL KEY
(0,K,S):: DISPLAY AT(24,1)::
 IF S=0 THEN 1590
1600 SUBEND
```

APOLLO

Ein Spiel in Ext. Basic, mit Joystick (1), in Anlehnung an das amerikanische Mondlandeprogramm, mit großer Detailtreue, bei dem vier Spielteile bewältigt werden müssen.

1. Bild:
Um die Erdumlaufbahn zu erzielen, müssen Sie bei –2 Sek. den Feuerknopf betätigen und den Joystick dann exakt nach oben drücken, bis die Saturn V den Bildschirm verläßt.

2. Bild:
Nun müssen Sie, nachdem Sie den durch Punkte gekennzeichneten Kurs gefolgt sind, im linken oberen Viertel des Mondes landen. Ein Verlassen des Kurses zeigt Ihr Bordcomputer durch Blinken des Punktes an, der die Position Ihres Raumschiffes markiert. Eine Explosion, die das vorzeitige Ende Ihrer Mission bedeuten würde, können Sie nur durch eine sofortige Kurskorrektur verhindern.

3. Bild:
Nach Ansteuern der Ebenen erreichen Sie Ihren Landeplatz; wobei die westliche Ebene einen höheren Punktemultiplikator bedeutet.

4. Bild:
Während Sie im Laufe des weiteren Sinkfluges eine der drei gekennzeichneten Landestellen ansteuern, können Sie Ihre Geschwindigkeit durch Betätigung des Feuerknopfes verringern. Bei einer geglückten Landung erhalten Sie auf der rechten Landefläche die höchste, auf der linken die zweithöchste Punktzahl. Eine weitere Saturn V stockt Ihre Flotte auf.

Anzeige:

Neben dem Schriftzug Apollo können Sie die Spielstufe ablesen, welche mit Apollo 11 beginnt und mit jeder Erhöhung eine geringere Kursabweichung und genauere Landung erfordert. Unter der Anzeige S=Sekunden, P=Punkte und H=Highscore wird Ihnen die zur Verfügung stehende Anzahl von Saturn V bzw. der Landefähren angezeigt. Ab dem zweiten Bild erscheint eine Tankanzeige. Ein leerer Tank führt zur Zerstörung Ihres Raumfahrzeuges. Ein Pilotton signalisiert Spielbeginn. Ein Verlassen des Bildschirms führt zu Punktabzug. Eine größere Zahl Raumfahrzeuge und eine höhere Spielstufe erhöhen den Punktemultiplikator.
Wartezeiten bei aufgebautem Bild können durch Feuer abgebrochen werden.
Wichtig! Alphalock entrasten.
Die Leerstellen in Zeile 150 mit CTRL und 1 eingeben.
Vor dem Laden eventuell CALL FILES (1) und NEW eingeben.

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *        APOLLO        *
13 ! *                      *
14 ! *     Copyright by     *
15 ! *                      *
16 ! *     Martin Winge     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *      Joystick        *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     12675 Bytes      *
28 ! *                      *
29 ! ************************
100 LE=5 :: SW,PU=0
110 RESTORE :: SL=0 :: GOSUB
 1230 :: CALL CHARSET :: CAL
L SCREEN(2)
120 CALL CHAR(92,"007E000000
00000000000000000000000000
00000000000000000000000000")
130 CALL CHAR(60,"4040404040
40E",64,"3C4299A1A199423C"):
: FOR C=2 TO 8 :: CALL COLOR
(C,16,1):: NEXT C :: DISPLAY
 AT(4,10):"A P O L L O"
140 FOR N=7 TO 20 STEP 2 ::
READ N$ :: DISPLAY AT(N,9):N
$ :: NEXT N
150 DATA ...@1985 BY,,MARTI
N WINGE,FRANZSTR.102,5800 HA
GEN 1,,..EX-BASIC
160 IF SW>0 THEN DISPLAY AT(
24,11):"FREISPIEL" :: LE=LE+
1
170 GOSUB 1300 :: CALL CLEAR
 :: CALL SCREEN(6):: CALL CH
AR(120,"",88,"81818181818181
81"):: CALL HCHAR(1,1,120,76
8):: SL=1
180 CALL CHAR(128,"010101030
3020303030303030303070400000
0808080808080808080808080C040"
)
190 CALL CHAR(132,"070707070
7070404070407070707070704C0C0C
0C0C0C04040C040C0C0C0C0C040"
)
200 CALL CHAR(136,"070707070
7070407040407070707F1F05C0C0C
0C0C0C040C04040C0C0C0E0F040"
)
210 CALL CHAR(140,"000000000
03F3F000000003F000000001F234
5F99BADC9AB9DA9CBAD99ABCD89"
)
220 CALL CHAR(96,"050D0F1F1B
0B0F0F0507020301010000404060E0
F070A0E06040C0808000000000")
230 CALL CHAR(100,"050D0B1D1
D0D0D0E07052507031311114060B
070706060E0C04048C080901010"
)
240 CALL CHAR(32,"FFFFFFFFFF
FFFFFF",35,"FFFCFEFCF8E0C080
",64,"0000007E")
250 CALL COLOR(1,6,1):: CALL
 VCHAR(1,1,32,72):: FOR I=4
TO 27 :: CALL VCHAR(1,I,32,2
7-I):: CALL VCHAR(28-I,I,35)
:: NEXT I
260 SE=15 :: GOSUB 1190 :: C
ALL SPRITE(#2,128,16,128,160
,#3,132,16,144,160,#4,136,16
,160,160):: CALL MAGNIFY(3)
270 DISPLAY AT(3,1)SIZE(4):"
S =@" :: CALL COLOR(12,4,3,1
,6,3):: CALL CHAR(120,"00001
01010000202")
280 CALL DELSPRITE(#5):: CAL
L MOTION(#2,0,0,#3,0,0,#4,0,
0):: GOSUB 1250
290 CALL SPRITE(#8,140,2,126
,168,#9,140,2,139,168,#10,14
0,2,152,168,#11,140,2,165,16
8)
300 CALL CHAR(40,"00000066FF
000000",41,"0000FFFFFFFFFFFF
",112,"0103070F1F3F7FFF",113
,"FFFEFCF8F0E0C080",114,"FFF
FFFFFFFFFFFFF")
310 CALL HCHAR(23,20,40,4)::
 CALL HCHAR(22,21,41,3):: CA
LL HCHAR(22,20,112):: CALL H
CHAR(22,24,114):: CALL HCHAR
(23,24,113)
320 CALL COLOR(2,8,2,11,2,3)
:: SE=SE-1 :: GOSUB 1290
330 IF SE<1 THEN 430 :: CALL
 KEY(1,K,S):: IF K<>18 THEN
320 :: IF SE=2 THEN 340 ELSE
 420
340 CALL SPRITE(#5,100,11,17
6,160)
350 FOR E=1 TO 0 STEP -1 ::
GOSUB 1200 :: CALL HCHAR(3,8
,E+48):: NEXT E :: SE=-1 ::
CALL HCHAR(3,6,32)
360 CALL PATTERN(#5,96):: CA
LL JOYST(1,XR,YR):: CALL PAT
TERN(#5,100)
370 SE=SE+1 :: GOSUB 1290 ::
 CALL PATTERN(#5,96)
380 CALL PATTERN(#5,100):: I
F YR<>4 THEN 420
390 CALL MOTION(#2,-YR,0,#3,
-YR,0,#4,-YR,0,#5,-YR,0):: C
ALL PATTERN(#5,96)
400 CALL COINC(#4,8,160,3,C)
```

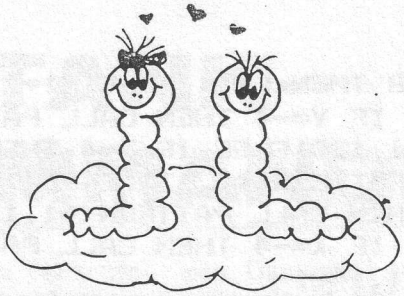
```
 :: IF C THEN GOTO 430
410 CALL PATTERN(#5,100):: P
U=PU+SE*(LE+SW):: GOSUB 1190
 :: GOTO 360
420 LE=LE-1 :: IF LE<1 THEN
100 ELSE 260
430 CALL CLEAR :: CALL CHAR(
32,""):: CALL SCREEN(2):: FO
R C=1 TO 3 :: CALL COLOR(C,2
,2):: NEXT C :: GOSUB 1230
440 FOR C=3 TO 8 :: CALL COL
OR(C,8,1):: NEXT C :: GOSUB
1190 :: KU=0 :: CALL MAGNIFY
(1):: RESTORE 450 :: FOR ER=
96 TO 111 :: READ ER$ :: CAL
L CHAR(ER,ER$):: NEXT ER
450 DATA 0F7FFFFFFFFFFFFF,FF
FFFFFFFFFFFFFF,01030307070F0
F0F,1F1F1F1F3F3F3F3F,3F3F3F3
F1F1F1F1F,0F0F0F0707030301,F
F7F3F1F0701,FFFFFFFFFFFF7F07
460 DATA FFFFFFFFFFFFFEE0,FF
FEFCF8E08,F0F0F0E0E0C0C080,F
CFCFCFCF8F8F8F8,F8F8F8F8FCFC
FCFC,FC787060201,000080E0F8F
CFEFF,F0FEFFFFFFFFFFFFFF
470 FOR ES=112 TO 143 :: REA
D ES$ :: CALL CHAR(ES,ES$)::
 NEXT ES
480 DATA 00FF663121,00000000
4A66FF,20E0711201623902,421C
A6C3388E673,00076E0073883CE,
401C70238EF84,C178324F221E70
01,10E0E4FE04F0BE83
490 DATA 0031F0873F6E381,039
E78D182107C3,364C7913C67C1C0
4,0,030F1F3F3F7F7B4D,4860703
F39190F03,C0F0D81C0C8696C2,E
AC24624FCF8F0C0
500 DATA 7EFCD09090803388,06
0F0F0F0F07070F,8060301D0F1F1
F1F,000000C0E0FCFEFE,0703030
1,1F3F3F1F1F0F0F03,C0F0FCFCF
8F8F0F0,030301010101
510 DATA FFFFFFFFFFFEFE78,E0
E0C08,0078FCF9D0C08,00004010
083C1882,0102010303030603,F8
F8F8B81C3C3C3C,3C3C1C3C38383
838,3030306060404
520 CALL CHAR(40,"000001071F
3F7FFF",41,"80C0C0E0E0F0F0F0
",42,"FFFFFFFFFFFFFFFF",37,"
52C3253342952942")
530 FOR EF=26 TO 29 :: CALL
VCHAR(18,EF,97,4):: NEXT EF
:: RESTORE 540 :: FOR EP=1 T
O 24 :: READ YE,XE,ZE :: CAL
L HCHAR(YE,XE,ZE):: NEXT EP
540 DATA 17,26,40,17,27,96,1
8,25,98,19,25,99,20,25,100,2
1,25,101,22,26,102,22,27,103
550 DATA 22,28,104,22,29,105
,21,30,106,20,30,107,19,30,1
08,18,30,41,17,29,110,17,28,
111,20,27,42
560 DATA 4,5,124,5,5,125,4,6
,126,5,6,127,16,29,95,4,3,95
,5,3,95,6,3,95
570 CALL HCHAR(15,25,95,5)::
 CALL HCHAR(7,3,95,5):: CALL
 HCHAR(3,3,95,4)
580 RESTORE 590 :: FOR EPS=2
1 TO 26 :: READ YE,XE :: CAL
L SPRITE(#EPS,117+EPS,2,YE*8
-7,XE*8-7):: NEXT EPS
590 DATA 17,27,17,29,18,25,1
9,30,20,30,21,30
600 CALL SPRITE(#27,109,2,22
*8-7,27*8-7)
610 RESTORE 620 :: FOR EPK=1
28 TO 137 :: READ Y,X :: CAL
L HCHAR(Y,X,EPK):: NEXT EPK
620 DATA 18,26,18,29,19,26,1
9,27,19,29,20,26,20,28,21,26
,21,27,21,28
630 RESTORE 640 :: FOR EWP=2
 TO 12 :: READ Y,X :: CALL S
PRITE(#EWP,110+EWP,2,Y,X)::
NEXT EWP
640 DATA 129,214,170,214,154
,212,158,220,152,202,159,209
,136,216,139,224,140,201,140
,209,165,222
650 CALL CHAR(36,"000000001"
,64,"0020702"):: FOR XB=2 TO
 31 STEP 6 :: CALL HCHAR(9+X
B/6,XB,36,6):: NEXT XB
660 RESTORE 670 :: FOR YB=1
TO 22 :: READ Y,X,W :: CALL
VCHAR(Y,X,36,W):: NEXT YB
670 DATA 16,28,1,16,27,1,16,
26,1,17,25,1,18,24,4,22,25,1
,23,26,1,23,27,1,23,28,1,23,
29,1
680 DATA 21,31,1,22,30,1,14,
31,7,3,2,6,2,3,1,2,4,1,2,5,1
690 DATA 2,6,1,3,7,3,6,5,1,6
,6,1,4,4,2
700 CALL COLOR(9,5,1,10,5,1,
1,9,1,2,3,1,12,11,1,13,3,5,1
4,3,5)
710 FOR C=2 TO 12 :: CALL CO
LOR(#C,15):: NEXT C :: FOR C
=21 TO 27 :: CALL COLOR(#C,3
):: NEXT C :: CALL STAR
720 IF LE<1 THEN 100 ELSE CA
LL SPRITE(#1,64,16,123,218):
: B=0 :: PT=1
730 GOSUB 1270 :: CALL COLOR
(#1,16):: PT=PT+.26 :: GOSUB
 1190 :: IF B=1 THEN 720 ELS
E GOSUB 1320
```

```
740 IF R=124 THEN PU=PU+(500
*(LE+SW)):: CALL MOTION(#1,0
,0):: GOTO 780 ELSE IF R=36
THEN KU=0 :: GOTO 770 ELSE I
F R>94 THEN KU=KU+2
750 KU=KU+1 :: PU=PU-1-SW ::
 CALL COLOR(#1,6):: IF KU<5-
SW THEN 770
760 GOSUB 1210 :: CALL DELSP
RITE(#1):: GOTO 720
770 CALL MOTION(#1,-Y/2,X/2)
:: GOTO 730
780 GOSUB 1190 :: GOSUB 1300
 :: GOSUB 1230 :: CALL CHAR(
60,"3C5A7E3C7E7E4281",116,"0
000E7"):: GOSUB 1190
790 CALL BERGE :: RESTORE 13
90 :: FOR B=96 TO 115 :: REA
D B$ :: CALL CHAR(B,B$):: NE
XT B
800 RESTORE 810 :: FOR MB=1
TO 38 :: READ Y,X,Z :: CALL
HCHAR(Y,X,Z):: NEXT MB
810 DATA 21,1,112,20,2,112,1
9,3,107,19,4,108,18,5,112,17
,6,112,16,7,112,16,8,97,16,9
,98,17,10,97
820 DATA 17,11,98,18,12,97,1
8,13,98,19,14,96,20,14,102,2
2,15,107,22,16,108,21,17,112
,20,18,109
830 DATA 21,13,103,22,13,104
840 DATA 19,18,110,18,19,109
,17,19,110,17,20,99,18,20,10
0,19,21,96,20,22,96,21,23,97
,21,24,98
850 DATA 21,27,112,20,28,109
,19,28,110,18,29,112,18,30,9
6,19,31,99,20,31,100,21,32,9
6
860 FOR MB=1 TO 15 :: READ Y
,X,W :: CALL HCHAR(Y,X,101,W
):: NEXT MB
870 DATA 17,7,3,18,6,6,19,5,
9,20,3,11,21,2,11,22,1,12,23
,1,32,24,1,32,19,19,2,20,19,
3
880 DATA 21,18,5,22,17,16,19
,29,2,20,29,2,21,28,4
890 FOR C=9 TO 14 :: CALL CO
LOR(C,11,1):: NEXT C :: CALL
 STAR
900 IF LE<1 THEN 100 ELSE CA
LL SPRITE(#1,60,15,16,16)::
B=0 :: PT=1
910 GOSUB 1270 :: IF B=1 THE
N 900 ELSE PT=PT+1.7 :: GOSU
B 1320 :: IF R<96 THEN 920 E
LSE GOSUB 1210 :: GOTO 900
920 IF YF>162+SW AND XF<114-
SW THEN PU=PU+(800*(LE+SW)):
: GOTO 930 ELSE IF YF>149+SW
 AND XF>192 THEN PU=PU+(400*
(LE+SW))ELSE 940
930 CALL DELSPRITE(#1):: GOS
UB 1190 :: GOSUB 1300 :: GOT
O 960
940 IF Y=0 AND X=0 THEN CALL
 MOTION(#1,6,5)ELSE CALL MOT
ION(#1,-Y*2,X*2)
950 GOTO 910
960 SL=4 :: GOSUB 1230 :: GO
SUB 1190 :: CALL MAGNIFY(3)
970 CALL CHAR(116,"000000000
0000000000000000000FA00000
0000000000000000000000000AF"
)
980 CALL CHAR(128,"00030F1B3
3331E0E3F333F12242142E004C8F
0D8CCCCF8F0FCBCFC4824844207"
)
990 CALL CHAR(132,"00030F1B3
3331E0E3F333F12252142E504C8F
0D8CCCCF8F0FCBCFC48548442A7"
)
1000 CALL CHAR(136,"00030F1B
B3F39E0E3F333F12242040E004C8
F0D8CCCCF8F0FCBCFC4824040207
")
1010 CALL CHAR(140,"00030F1B
33331E0E3F333F12242040E004C8
F0D8CDCEF9F0FCBCFC4824040207
")
1020 CALL CHAR(124,"00030F1B
33331E0E3F333F12242040E004C8
F0D8CCCCF8F0FCBCFC4824040207
")
1030 CALL CHAR(120,"0820E504
203208829A2929380083123000014
32280ECC26202A82660090B801")
1040 CALL BERGE :: RESTORE 1
390 :: FOR B=96 TO 114 :: RE
AD B$ :: CALL CHAR(B,B$):: N
EXT B
1050 CALL POSB :: RESTORE 14
40 :: FOR BP=1 TO 72 :: READ
 Y,X,Z :: CALL HCHAR(Y,X,Z):
: NEXT BP
1060 CALL HCHAR(24,1,101,32)
:: FOR C=9 TO 11 :: CALL COL
OR(C,11,2):: NEXT C
1070 CALL SPRITE(#26,116,5,7
6,144,#27,116,5,164,40,#28,1
16,5,164,218):: GOSUB 1190 :
: GOSUB 1240 :: CALL STAR
1080 IF LE<1 THEN 100 ELSE C
ALL SPRITE(#1,124,15,16,60,0
,0):: B=0 :: PT=1
1090 G1=4 :: GOSUB 1270 :: I
F B=1 THEN 1080 ELSE PT=PT+2
.3 :: IF X=0 AND Y=0 THEN 11
20 ELSE CALL KEY(1,K,S):: IF
```

```
 K=18 THEN G1=1 ELSE G1=2
1100 IF Y=-4 THEN CALL PATTE
RN(#1,128)ELSE IF Y=4 THEN C
ALL PATTERN(#1,132)ELSE IF X
=4 THEN CALL PATTERN(#1,136)
ELSE IF X=-4 THEN CALL PATTE
RN(#1,140)
1110 CALL COINC(#1,#26,9-2*S
W,C1):: CALL COINC(#1,#27,7-
2*SW,C2):: CALL COINC(#1,#28
,6-2*SW,C3):: IF C1 OR C2 OR
 C3 THEN 1160
1120 IF XF/8>237 THEN 1080 E
LSE CALL GCHAR((YF/8)+1,(XF/
8)+1,R1):: CALL GCHAR((YF/8)
+1,(XF/8),R2):: CALL GCHAR((
YF/8)+2,(XF/8)+2,R3)
1130 IF R1<96 AND R2<96 AND
R3<96 THEN 1140 ELSE GOSUB 1
210 :: GOTO 1080
1140 IF G1>3 THEN CALL MOTIO
N(#1,G1,G1+1)ELSE CALL MOTIO
N(#1,-Y*G1,X*G1)
1150 GOTO 1090
1160 CALL MOTION(#1,0,0):: C
ALL PATTERN(#1,128):: CALL P
OSITION(#1,YP,XP):: CALL SOU
ND(800,440,4)
1170 IF YP<96 THEN PU=PU+(50
0*(LE+SW))ELSE IF XP<80 THEN
 PU=PU+(1000*(LE+SW))ELSE IF
 XP>206 THEN PU=PU+(1500*(LE
+SW))
1180 GOSUB 1190 :: SW=SW+1 :
: GOSUB 1300 :: GOTO 110
1190 DISPLAY AT(11,1)SIZE(7)
:"P";PU :: RETURN
1200 CALL SOUND(818,44733,30
):: RETURN
1210 CALL MOTION(#1,0,0):: I
F SL=4 THEN CALL PATTERN(#1,
120)ELSE CALL SPRITE(#1,37,1
1,YF,XF)
1220 LE=LE-1 :: FOR I=1 TO 2
0 :: CALL SOUND(-40,-6,I)::
NEXT I :: FOR I=1 TO 20 :: N
EXT I :: CALL DELSPRITE(#1):
: GOTO 1250
1230 CALL DELSPRITE(ALL):: C
ALL CLEAR :: FOR C=9 TO 14 :
: CALL COLOR(C,2,2):: NEXT C
 :: IF SL=0 THEN RETURN ELSE
 GOTO 1240
1240 CALL VCHAR(1,31,88,9)::
 CALL SPRITE(#15,92,3,1,241)
1250 DISPLAY AT(15,1)SIZE(3)
:"<";LE :: IF PU>H THEN H=PU
1260 DISPLAY AT(13,1)SIZE(7)
:"H";H :: DISPLAY AT(1,9)SIZ
E(13):"APOLLO";SW+11 :: RETU
RN
1270 CALL JOYST(1,X,Y):: CAL
L LOCATE(#15,INT(PT),241)::
CALL POSITION(#1,YF,XF):: IF
 PT>71 THEN B=1 :: GOTO 1210
1280 IF XF>5 AND XF<247 AND
YF<187 AND YF>3 THEN RETURN
ELSE CALL MOTION(#1,0,0):: P
U=PU-500 :: CALL SOUND(100,2
20,2):: B=1 :: GOTO 1190
1290 E=INT(SE/10):: F=SE-E*1
0 :: CALL HCHAR(3,7,E+48)::
CALL HCHAR(3,8,F+48):: GOTO
1200
1300 W=0
1310 W=W+1 :: CALL KEY(1,K,S
):: IF K=18 OR W>350 THEN RE
TURN ELSE 1310
1320 CALL GCHAR(INT(YF/8)+1,
INT(XF/8)+1,R):: RETURN
1330 SUB STAR :: SS=91 :: CA
LL CHAR(91,"8",58,"000101")
1340 FOR I=1 TO 15
1350 RANDOMIZE :: YS=INT(24*
RND)+1 :: XS=INT(32*RND)+1 :
: CALL GCHAR(YS,XS,PS):: IF
PS<>32 THEN 1350 :: CALL HCH
AR(YS,XS,SS)
1360 NEXT I :: IF SS=91 THEN
 SS=58 :: GOTO 1340 ELSE IF
SS=58 THEN SS=36 :: GOTO 134
0 ELSE IF SS=36 THEN 1370
1370 CALL SOUND(100,932,2)::
 SUBEND
1380 END
1390 SUB BERGE
1400 DATA 80C0E0F0F8FCFEFF,C
0F0FCFFFFFFFFFF,00000000C0F0
FCFF,8080C0C0E0E0F0F0,F8F8FC
FCFEFEFFFF,FFFFFFFFFFFFFFFF,
FFFEFCF8F0E0C080,FFFFFEFEFCF
CF8F8
1410 DATA F0F0E0E0C0C08080,F
FFCF0C0,FFFFFFFFFFCF0C0,000
00000030F3FFF,030F3FFFFFFFFF
FF,1F1F3F3F7F7FFFFF,01010303
07070F0F,FF7F3F1F0F070301
1420 DATA 0103070F1F3F7FFF,0
F0F070703030101,FFFF7F7F3F3F
1F1F
1430 SUBEND
1440 SUB POSB
1450 DATA 1,1,96,2,2,96,3,3,
97,3,4,98,4,5,97,4,6,98,5,7,
96,6,8,96,7,9,99,8,9,100
1460 DATA 10,10,99,11,10,100
,12,10,101,13,10,102,14,9,10
3,15,9,104,16,8,105,16,7,106
1470 DATA 17,4,102,18,3,102,
19,2,101,20,3,99,21,3,100,22
,4,96,23,5,101,23,6,101
1480 DATA 23,7,101,23,8,101,
```

```
22,9,107,22,10,108,21,11,109
,20,11,110,19,12,107,19,13,1
08
1490 DATA 18,14,112,17,15,10
9,16,15,110,15,15,113,14,15,
114,13,15,112,12,16,107,12,1
7,108
1500 DATA 12,22,99,13,22,100
,14,23,96,15,24,99,16,24,100
,17,25,99,18,25,104,19,25,99
,20,25,100
1510 DATA 21,25,102,22,25,97
,22,26,98,22,30,112,21,31,10
9,20,31,110,19,31,111,18,30,
113,17,30,114
1520 DATA 16,29,113,15,29,11
4,14,28,111,13,27,111,12,26,
111,11,26,112,10,27,107,10,2
8,98,10,29,107
1530 DATA 10,30,98,11,31,96,
11,32,107
1540 RESTORE 1560
1550 FOR V=1 TO 35 :: READ Y
,X,W :: CALL VCHAR(Y,X,101,W
):: NEXT V :: CALL HCHAR(23,
26,101,5)
1560 DATA 2,1,22,3,2,21,4,3,
14,22,3,2,4,4,13,23,4,1,5,5,
12,5,6,12,6,7,10,7,8,9,9,9,5
1570 DATA 23,9,1,23,10,1,22,
11,2,20,12,4,20,13,4,19,14,5
,18,15,6,13,16,11,13,17,11,1
2,18,12,12,19,12
1580 DATA 12,20,12,12,21,12,
14,22,10,15,23,9,17,24,7,23,
25,1,11,27,2,11,28,3,11,29,4
,11,30,6
1590 DATA 12,31,7,22,31,2,12
,32,13
1600 SUBEND
```

LISTINGS



MISSION ZULU

Als Commander eines Raumschiffes ist es Ihre Aufgabe, Asteroidenfelder zu räumen. Dabei müssen Sie darauf achten, daß Sie immer genügend Treibstoff und Sauerstoff zur Verfügung haben.
Nach dem Titelbild, das erst durch Drücken einer beliebigen Taste verschwindet, können Sie wählen, mit welchen Tasten das Raumschiff gesteuert werden soll (90 Grad Links- bzw. Rechtsdrehung und Feuern). Nun wird das Spielfeld aufgebaut. Oberhalb des Spielfelds sind die zur Verfügung stehenden Raumschiffe und der Highscore verzeichnet. Unterhalb des Spielfeldes sind der Punktestand, der Benzinvorrat und der Luftvorrat abzulesen. Der Benzinvorrat nimmt bei jedem Schuß und bei jeder Drehung des Raumschiffes ab. Er wird durch Abschuß der Benzintanks wieder erhöht. Der Luftvorrat nimmt bei jeder Bewegung des Raumschiffes ab und wird durch Überfahren der Luftblasen aufgefüllt. Asteroiden müssen abgeschossen werden. Das Raumschiff bewegt sich ständig vorwärts, wenn nicht eine der Steuertasten gedrückt wurde. Falls der Luftvorrat zu Ende ist, explodiert das Raumschiff, bei leerem Tank ist es nicht mehr steuerbar. Der Schwierigkeitsgrad wird von Bild zu Bild erhöht. Nach zwei erfolgreich durchlaufenen Bildern gibt es jeweils ein neues Raumschiff. Eine Spielpause kann mit der ENTER-Taste eingeschaltet werden. Diese Taste kann daher nicht zur Steuerung des Raumschiffes benutzt werden!

BÖRSE

Neue Module EX BASIC mit deutschem Handbuch DM 120; Statistics Germ. Mod. DM 35; Buchungsjournal Mod. DM 35; Othello Modul DM 30; MINUS MIS. u. ALIEN ADDIT. auch als Selbstbaugehäuse DM 7,50. Jürgen Reimer 040/2994609.

Verkaufe wegen Systemaufgabe folgende Module: MASH, TI-Invaders, Munch-Man, Sahtzee, Othello, Wumpusjagd! Dirk Kronshage, Akazienstraße 24, 4902 Bad SAlzuflen oder Telefon 05222/13837.

TI 99/4A + Exbasic + neuer Recorder + Kabel + Listings + Spiele + 5 Bücher VB 400 DM
Michael Laegel, Königstforststr. 36, 5000 Köln 91, 0221/84 11 20

TI99/4A+Ext-Basic 360 DM (nur zusammen); Module Statistik, Report-Generator, Datenverwaltung je 75 DM, Minus Mission, Alien Addition je 25 DM, Rec-Kabel 15 DM dt. Handbuch Ext-Basic 30 DM Klaus Lindner, Weilerstr. 7 5303 Bornheim 4, Tel. 02227/7040

Verk.: TI-99/4A + X-Ba + Rec. + Kabel + Parsec + 2 Joyst. + Lit. DM 320,--.
C. Berski, Oldenkotterstr. 65, 4426 Vreden.

Suche ext. 32-K. Erweiterung Epson-Drucker RX 80 (nagelneu/originalverpackt), Preis VB. Datenverw. + Analyse 50,-- DM. Ralf Theile (0209) 784272.

Verk. TI 99/4 A + Ex.-Basic + 4 Module + Rec. Kabel + 2 Joyst. + viel Lit. u. Bücher + sehr viel Softw. Preis: 370 VB. Tel.: 06103/24605.

TI 99/4A + Video-Game 12, Blackjack/Poker, Parsec, Alpiner. Alle Preise VS. Verk.: 06441/74830.

Verkaufe TI 99/4 A + orig. Ext.-Basic Modul mit engl. u. dt. Handbuch + Rec. Kabel + Joyst. m. Adapter + ca. 100 Spielprogramme. Alles nur 18 Monate alt für DM 420. D. Stüber, Tel.: 089/ 756207.

Winterschlußverkauf! Mini-Assembler (Radix) komplett mit Software für sympathische 80,-- DM bei: Kay: 040/5254608.

LISTINGS

```
10 REM**********************
11 REM*   MISSION ZULU     *
12 REM*                    *
13 REM*   Copyright by     *
14 REM*                    *
15 REM*  Andreas kaluza    *
16 REM*                    *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                    *
23 REM* Speicherbelegung *
24 REM*    8448 Bytes     *
25 REM**********************
26 REM
100 SC1=750
110 SC=0
120 DIM P(20)
130 GOSUB 4100
140 FOR I=2 TO 8
150 CALL COLOR(I,2,1)
160 NEXT I
170 CALL SCREEN(7)
180 CALL CLEAR
190 A$="TASTE ZUM LINKS DREH
EN:_"
200 A=9
210 B=6
220 GOSUB 1630
230 CALL KEY(0,K,S)
240 IF S=0 THEN 230
250 FOR I=1 TO 5
260 Z$(I)=""
270 NEXT I
280 IF (K<32)+(K>90)THEN 230
290 CALL HCHAR(9,29,K)
300 A$="TASTE ZUM RECHTS DRE
HEN:_"
310 A=12
320 B=6
330 GOSUB 1630
340 DL=K
350 CALL KEY(0,K,S)
360 IF S=0 THEN 350
370 IF (K<32)+(K>90)+(K=DL)T
HEN 350
380 CALL HCHAR(12,30,K)
390 A$="TASTE ZUM SCHIESSEN:
 _"
400 A=15
410 B=6
420 GOSUB 1630
430 DR=K
440 CALL KEY(0,K,S)
450 IF S=0 THEN 440
460 IF (K<32)+(K>90)+(K=DL)+
(K=DR)THEN 440
470 CALL HCHAR(15,26,K)
480 SCH=K
490 A$="TASTATURBELEGUNG RIC
HTIG?"
500 A=18
510 B=6
520 GOSUB 1630
530 CALL KEY(0,K,S)
540 IF S=0 THEN 530
550 IF K=78 THEN 100
560 IF K=74 THEN 570 ELSE 53
0
570 Y(1)=-1
580 Y(2)=0
590 Y(3)=1
600 Y(4)=0
610 X(1)=0
620 X(2)=1
630 X(3)=0
640 X(4)=-1
650 CALL CHAR(96,"18183C66FF
FFDB81")
660 CALL CHAR(121,"E7E7C3181
8C3E7E7")
670 CALL CHAR(97,"F0783C7777
3C78F")
680 CALL CHAR(98,"81DBFFFF66
3C1818")
690 CALL CHAR(101,"288A532C9
97414A2")
700 CALL CHAR(99,"0F1E3CEEEE
3C1E0F")
710 CALL CHAR(100,"000000181
8")
720 CALL CHAR(104,"000000FFF
F")
730 CALL CHAR(105,"3C4291919
19D423C")
740 CALL CHAR(112,"000000FFF
F")
750 CALL CHAR(113,"FF81DDD1D
9D191FF")
760 CALL CHAR(120,"1C2248A2A
A89720C")
770 Z=96
780 FOR I=1 TO 4
790 HCH(I)=Z
800 Z=Z+1
810 NEXT I
820 CALL COLOR(9,15,2)
830 CALL COLOR(10,5,2)
840 CALL COLOR(11,11,2)
850 CALL COLOR(12,7,2)
860 ANZ=40
870 CALL CLEAR
880 CALL SCREEN(2)
890 FOR I=2 TO 8
900 CALL COLOR(I,15,2)
910 NEXT I
920 SH=3
930 CALL HCHAR(3,3,121,30)
940 CALL VCHAR(3,32,121,22)
950 CALL HCHAR(24,3,121,30)
960 CALL VCHAR(22,23,121,2)
970 CALL HCHAR(21,3,121,30)
980 CALL VCHAR(3,3,121,22)
```

LISTINGS

```
990 L=1
1000 AR=70
1010 F=28
1020 RANDOMIZE
1030 CALL HCHAR(23,9,104,14)
1040 CALL HCHAR(22,9,112,14)
1050 A$="FUEL:"
1060 A=22
1070 B=4
1080 GOSUB 1630
1090 A$="AIR :"
1100 A=23
1110 B=4
1120 GOSUB 1630
1130 A$="SCORE:"
1140 A=22
1150 B=25
1160 GOSUB 1630
1170 A$="SHIPS:"&STR$(SH)
1180 A=2
1190 B=3
1200 GOSUB 1630
1210 A$="HIGHSCORE:"&STR$(HI
)
1220 A=2
1230 B=14
1240 GOSUB 1630
1250 A$=STR$(SC)
1260 A=23
1270 B=25
1280 GOSUB 1630
1290 FOR I=1 TO ANZ
1300 X1=INT(RND*28)+4
1310 Y1=INT(RND*17)+4
1320 CALL GCHAR(Y1,X1,G)
1330 IF G<>32 THEN 1300
1340 CALL HCHAR(Y1,X1,120)
1350 NEXT I
1360 FOR I=1 TO 20
1370 X1=INT(RND*28)+4
1380 Y1=INT(RND*17)+4
1390 CALL GCHAR(Y1,X1,G)
1400 IF G<>32 THEN 1370
1410 CALL HCHAR(Y1,X1,113)
1420 NEXT I
1430 FOR I=1 TO 15
1440 X1=INT(RND*28)+4
1450 Y1=INT(RND*17)+4
1460 CALL GCHAR(Y1,X1,G)
1470 IF G<>32 THEN 1440
1480 CALL HCHAR(Y1,X1,105)
1490 NEXT I
1500 X2=INT(RND*28)+4
1510 Y2=INT(RND*17)+4
1520 CALL GCHAR(Y2,X2,G)
1530 IF G<>32 THEN 1500
1540 CALL HCHAR(Y2,X2,HCH(L)
)
1550 FOR I=1 TO 21 STEP 3
1560 CALL HCHAR(Y2,X2,32)
1570 CALL SOUND(150,700-I*5,
I)
1580 CALL HCHAR(Y2,X2,HCH(L)
)
1590 CALL SOUND(150,300+I*25
,I)
1600 NEXT I
1610 GOSUB 1680
1620 GOTO 1610
1630 FOR I=1 TO LEN(A$)
1640 CALL HCHAR(A,(B+I)-1,AS
C(SEG$(A$,I,1)))
1650 NEXT I
1660 CALL SOUND(100,1000,2)
1670 RETURN
1680 CALL KEY(0,K,S)
1690 IF K=DL THEN 1990
1700 IF K=DR THEN 2070
1710 IF K=13 THEN 2630
1720 IF K=SCH THEN 2230
1730 CALL HCHAR(Y2,X2,32)
1740 Y2=Y2+Y(L)
1750 X2=X2+X(L)
1760 CALL GCHAR(Y2,X2,G)
1770 IF G=32 THEN 1800
1780 IF G=105 THEN 1900
1790 IF (G=120)+(G=113)+(G=1
21)THEN 2890
1800 REM
1810 CALL HCHAR(Y2,X2,HCH(L)
)
1820 AR=AR-1
1830 CALL HCHAR(23,9,32,14)
1840 IF INT(AR/5)<15 THEN 18
70
1850 CALL HCHAR(23,9,104,14)
1860 RETURN
1870 CALL HCHAR(23,9,104,INT
(AR/5))
1880 IF AR=0 THEN 2890
1890 RETURN
1900 AR=AR+10
1910 IF INT(AR/5)<15 THEN 19
40
1920 CALL HCHAR(23,9,104,14)
1930 RETURN
1940 CALL HCHAR(23,9,104,INT
(AR/5))
1950 FOR I=2000 TO 400 STEP
-400
1960 CALL SOUND(-100,I,4)
1970 NEXT I
1980 RETURN
1990 IF F=0 THEN 1730
2000 L=L-1
2010 IF L=0 THEN 2050
2020 CALL HCHAR(Y2,X2,HCH(L)
)
2030 GOSUB 2150
2040 GOTO 1820
2050 L=4
2060 GOTO 2020
```

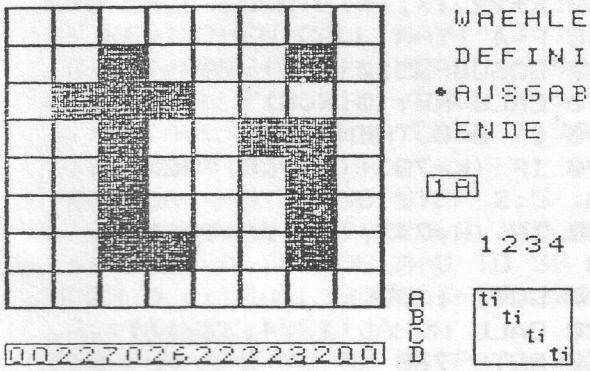
LISTINGS

```
2070 IF F=0 THEN 1730
2080 L=L+1
2090 IF L=5 THEN 2130
2100 CALL HCHAR(Y2,X2,HCH(L)
)
2110 GOSUB 2150
2120 GOTO 1820
2130 L=1
2140 GOTO 2100
2150 F=F-1
2160 CALL HCHAR(22,9,32,14)
2170 IF INT(F/2)<15 THEN 220
0
2180 CALL HCHAR(22,9,112,14)
2190 RETURN
2200 CALL HCHAR(22,9,112,INT
(F/2))
2210 CALL SOUND(100,110,2,-5
,2)
2220 RETURN
2230 SO=0
2240 C=Y2
2250 D=X2
2260 CALL SOUND(50,-3,SO)
2270 SO=SO+1
2280 C=C+Y(L)
2290 D=D+X(L)
2300 CALL GCHAR(C,D,G)
2310 IF (G=120)+(G=105)THEN
2370
2320 IF G=121 THEN 2430
2330 IF G=113 THEN 2450
2340 CALL HCHAR(C,D,100)
2350 CALL HCHAR(C,D,32)
2360 GOTO 2260
2370 CALL HCHAR(C,D,101)
2380 FOR I=1 TO 30
2390 CALL SOUND(-100,-5.9,I)
2400 NEXT I
2410 CALL SOUND(500,-7.3,5)
2420 CALL HCHAR(C,D,32)
2430 GOSUB 2150
2440 GOTO 1820
2450 CALL HCHAR(C,D,101)
2460 FOR I=1 TO 30 STEP 3
2470 CALL SOUND(-50,-1.5,I)
2480 NEXT I
2490 CALL SOUND(-600,110,6,-
7.5,3)
2500 CALL HCHAR(C,D,32)
2510 F=F+1
2520 IF INT(F/2)<15 THEN 255
0
2530 CALL HCHAR(22,9,112,14)
2540 GOTO 2560
2550 CALL HCHAR(22,9,112,INT
(F/2))
2560 A=23
2570 B=25
2580 SC=SC+50
2590 A$=STR$(SC)
2600 GOSUB 1630
2610 IF SC=SC1 THEN 2800
2620 GOTO 1820
2630 FOR I=15 TO 19
2640 CALL GCHAR(12,I,G)
2650 P(I)=G
2660 NEXT I
2670 A$="BREAK"
2680 A=12
2690 B=15
2700 GOSUB 1630
2710 CALL KEY(0,K,S)
2720 IF S=0 THEN 2710
2730 IF K<>13 THEN 2710
2740 FOR I=15 TO 19
2750 CALL HCHAR(12,I,P(I))
2760 NEXT I
2770 FOR I=1 TO 100
2780 NEXT I
2790 RETURN
2800 CALL VCHAR(1,26,32,384)
2810 CALL HCHAR(1,1,32,800)
2820 ANZ=ANZ+15
2830 SC1=SC1+1000
2840 SC=SC+250
2850 D=D+1
2860 IF D/2<>INT(D/2)THEN 28
80
2870 SH=SH+1
2880 GOTO 930
2890 FOR I=3 TO 16
2900 CALL SOUND(-100,-5.9,I)
2910 CALL SCREEN(I)
2920 NEXT I
2930 CALL SOUND(500,-7.5,5)
2940 CALL SCREEN(2)
2950 SH=SH-1
2960 IF SH=0 THEN 3110
2970 CALL HCHAR(Y2,X2,32)
2980 A$=STR$(SH)
2990 CALL HCHAR(3,3,121,30)
3000 CALL HCHAR(21,3,121,30)
3010 CALL VCHAR(3,3,121,20)
3020 CALL VCHAR(3,32,121,20)
3030 A=2
3040 B=9
3050 GOSUB 1630
3060 AR=70
3070 F=28
3080 CALL HCHAR(23,9,104,14)
3090 CALL HCHAR(22,9,112,14)
3100 GOTO 1500
3110 IF SC<HI THEN 3130
3120 HI=SC
3130 CALL CLEAR
3140 A$="DEIN PUNKTESTAND"
3150 B=9
3160 A=2
3170 GOSUB 1630
3180 CALL CHAR(102,"FFFFFFFF
FFFFFFFF")
```

```
3190 C$=STR$(SC)
3200 FOR I=1 TO LEN(C$)
3210 D$=SEG$(C$,I,1)
3220 C=VAL(D$)
3230 GOSUB 3440
3240 NEXT I
3250 A=5
3260 FOR O=1 TO 5
3270 A$=Z$(O)
3280 A=A+1
3290 B=9
3300 GOSUB 1630
3310 NEXT O
3320 FOR I=1 TO 2000
3330 NEXT I
3340 A$="NOCH EIN SPIEL ?"
3350 A=20
3360 B=9
3370 GOSUB 1630
3380 CALL KEY(0,K,S)
3390 IF S=0 THEN 3380
3400 IF K=74 THEN 100
3410 IF K=78 THEN 3430
3420 GOTO 3380
3430 END
3440 IF C=0 THEN 3500
3450 ON C GOSUB 3560,3620,36
80,3740,3800,3860,3920,3980,
4040
3460 FOR J=1 TO 5
3470 Z$(J)=Z$(J)&L$(J)
3480 NEXT J
3490 RETURN
3500 L$(1)="fff.."
3510 L$(2)="f f.."
3520 L$(3)="f f.."
3530 L$(4)="f f.."
3540 L$(5)="fff.."
3550 GOTO 3460
3560 L$(1)=" f.."
3570 L$(2)="ff.."
3580 L$(3)=" f.."
3590 L$(4)=" f.."
3600 L$(5)=" f.."
3610 RETURN
3620 L$(1)="fff.."
3630 L$(2)="..f.."
3640 L$(3)="fff.."
3650 L$(4)="f...."
3660 L$(5)="fff.."
3670 RETURN
3680 L$(1)="fff.."
3690 L$(2)="..f.."
3700 L$(3)="fff.."
3710 L$(4)="..f.."
3720 L$(5)="fff.."
3730 RETURN
3740 L$(1)="f f.."
3750 L$(2)="f f.."
3760 L$(3)="fff.."
3770 L$(4)="..f.."
3780 L$(5)="..f.."
3790 RETURN
3800 L$(1)="fff.."
3810 L$(2)="f...."
3820 L$(3)="fff.."
3830 L$(4)="..f.."
3840 L$(5)="fff.."
3850 RETURN
3860 L$(1)="fff.."
3870 L$(2)="f...."
3880 L$(3)="fff.."
3890 L$(4)="f f.."
3900 L$(5)="fff.."
3910 RETURN
3920 L$(1)="fff.."
3930 L$(2)="f f.."
3940 L$(3)="..f.."
3950 L$(4)="..f.."
3960 L$(5)="..f.."
3970 RETURN
3980 L$(1)="fff.."
3990 L$(2)="f f.."
4000 L$(3)="fff.."
4010 L$(4)="f f.."
4020 L$(5)="fff.."
4030 RETURN
4040 L$(1)="fff.."
4050 L$(2)="f f.."
4060 L$(3)="fff.."
4070 L$(4)="..f.."
4080 L$(5)="fff.."
4090 RETURN
4100 CALL CLEAR
4110 CALL SCREEN(2)
4120 FOR I=2 TO 9
4130 CALL COLOR(I,15,2)
4140 NEXT I
4150 CALL CHAR(102,"FFFFFFFF
FFFFFFFF")
4160 CALL VCHAR(1,3,102,24)
4170 CALL VCHAR(1,32,102,24)
4180 CALL HCHAR(1,3,102,30)
4190 CALL HCHAR(24,3,102,30)
4200 CALL CHAR(40,"3C4299A1A
199423C")
4210 A$="FALCON SOFTWARE PRE
SENTS :"
4220 A=3
4230 B=5
4240 GOSUB 1630
4250 A$="MISSION ZULU"
4260 A=8
4270 B=12
4280 GOSUB 1630
4290 A$="( BY ANDREAS KALUZA
"
4300 A=12
4310 B=8
4320 GOSUB 1630
4330 A$="GEBR. KALUZA"
4340 A=17
```

```
4350 B=5
4360 GOSUB 1630
4370 A$="ZUR MAERK 14"
4380 A=19
4390 B=5
4400 GOSUB 1630
4410 A$="3530 WARBURG"
4420 A=21
4430 B=5
4440 GOSUB 1630
4450 CALL HCHAR(6,10,42,16)
4460 CALL HCHAR(10,10,42,16)
4470 CALL VCHAR(6,10,42,5)
4480 CALL VCHAR(6,25,42,5)
4490 CALL KEY(0,K,S)
4500 IF S=0 THEN 4490
4510 RETURN
```

CHAR



Nur Konsole und Joystick brauchen Sie, um mit diesem Programm ganz einfach Character zu erstellen. Außerdem können bis zu 16 Character in einem 4x4 Raster dargestellt werden, was beim Erstellen größerer Gegenstände für Grafiken sehr hilfreich ist. Daß dieses überaus nützliche Programm den HEX-Code der erstellten Character überaus schnell (!) ausgibt, ist selbstverständlich.
Die einzelnen Punkte des Menüs werden durch einen weißen Punkt per Joystick angesteuert und durch Drücken der Feuertaste angewählt.

Definieren:
Nach Anwählen dieser Option erscheint im Zeichenfeld (8x8 Raster) ein Punkt, der mit dem Joystick gesteuert werden kann. Drücken des Feuerknopfes setzt ein Pixel, falls auf dem Feld ein Pixel war, so wird es gelöscht. Hat man den Character beendet, führt man den Punkt über den Rand des Zeichenfeldes hinaus und ist damit wieder im Menü. Als nächstes wählt man die Option

Wählen:
Damit kommt man zum 4x4 Raster, das mit den Koordinaten A–D und 1–4 versehen ist. Mit dem Joystick kann man die beiden weißen Punkte verschieben bis das Feld gewählt ist, in das man den zuvor definierten Character setzen will. Das Setzen des Characters erfolgt wieder mit der Feuertaste, woraufhin der Character im 4x4 Raster abgebildet ist. Man ist nun wieder im Menü und kann denselben Character nochmal in das 4x4 Raster

Bitte lesen Sie weiter auf Seite 48

LISTINGS

```
10 REM*********************
11 REM*       CHAR        *
12 REM*                   *
13 REM*   Copyright by    *
14 REM*                   *
15 REM*  Andreas Kaluza   *
16 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*     Joystick      *
20 REM*                   *
23 REM* Speicherbelegung *
24 REM*    5756 Bytes     *
25 REM*********************
26 REM
100 CALL CLEAR
110 DIM G(4,4),DFE$(4,4),Q(8
,4),W(8,4)
120 CALL COLOR(11,2,15)
130 CALL COLOR(2,2,15)
140 CALL COLOR(8,2,15)
150 CALL COLOR(1,16,1)
160 CALL SCREEN(6)
170 FOR I=1 TO 23
180 READ A,A$
190 CALL CHAR(A,A$)
200 NEXT I
210 FOR I=1 TO 28
220 READ A,B,C,D
230 CALL HCHAR(A,B,C,D)
240 NEXT I
250 CALL VCHAR(4,3,102,16)
260 CALL VCHAR(4,20,103,16)
270 CALL VCHAR(19,23,102,4)
280 CALL VCHAR(19,28,103,4)
290 GOTO 400
300 FOR J=4 TO 18 STEP 2
310 FOR I=4 TO 18 STEP 2
320 CALL HCHAR(I,J,112)
330 CALL HCHAR(I,J+1,113)
340 CALL HCHAR(I+1,J,114)
350 CALL HCHAR(I+1,J+1,115)
360 NEXT I
370 NEXT J
380 EFD$="0000000000000000"
390 GOTO 660
400 GOSUB 300
410 FOR J=1 TO 9
420 READ A$,E,F
430 GOSUB 2510
440 NEXT J
450 FOR I=120 TO 135
460 CALL CHAR(I,"")
470 NEXT I
480 L=119
490 FOR I=19 TO 22
500 FOR J=24 TO 27
510 L=L+1
520 G(J-23,I-18)=L
530 CALL HCHAR(I,J,L)
540 NEXT J
```

```
550 NEXT I
560 FOR I=1 TO 4
570 FOR J=1 TO 4
580 DFE$(J,I)="0000000000000
000"
590 NEXT J
600 NEXT I
610 A$="1A"
620 E=13
630 F=22
640 GOSUB 2510
650 GOTO 730
660 FOR I=1 TO 8
670 FOR J=1 TO 4
680 Q(I,J)=1
690 W(I,J)=1
700 NEXT J
710 NEXT I
720 RETURN
730 T=112
740 Z=4
750 CALL HCHAR(Z,22,38)
760 CALL JOYST(1,X,Y)
770 CALL KEY(1,K,S)
780 IF K=18 THEN 890
790 IF K<>14 THEN 810
800 GOSUB 300
810 IF Y<>0 THEN 820 ELSE 76
0
820 CALL HCHAR(Z,22,32)
830 Z=Z+Y/-2
840 IF (Z<4)+(Z>10)THEN 870
850 CALL HCHAR(Z,22,38)
860 GOTO 760
870 Z=Z-Y/-2
880 GOTO 850
890 ON Z-3 GOTO 920,900,1470
,900,920,900,1330
900 GOTO 760
910 REM **WAEHLEN**
920 A$="1A"
930 E=13
940 F=22
950 GOSUB 2510
960 A=24
970 B=19
980 CALL HCHAR(B,22,38)
990 CALL HCHAR(17,A,38)
1000 CALL JOYST(1,X,Y)
1010 CALL KEY(1,K,S)
1020 IF K=18 THEN 1180
1030 IF X<>0 THEN 1120
1040 IF Y<>0 THEN 1060
1050 GOTO 1000
1060 IF (B-Y/4<19)+(B-Y/4>22
)THEN 1000
1070 CALL HCHAR(B,22,32)
1080 B=B-Y/4
1090 CALL HCHAR(B,22,38)
1100 CALL HCHAR(13,23,B+46)
1110 GOTO 1000
```

LISTINGS

```
1120 IF (A+X/4<24)+(A+X/4>27
)THEN 1000
1130 CALL HCHAR(17,A,32)
1140 A=A+X/4
1150 CALL HCHAR(17,A,38)
1160 CALL HCHAR(13,22,A+25)
1170 GOTO 1000
1180 IF Z=8 THEN 1250
1190 DFE$(A-23,B-18)=EFD$
1200 CALL CHAR(G(A-23,B-18),
DFE$(A-23,B-18))
1210 CALL HCHAR(B,22,32)
1220 CALL HCHAR(17,A,32)
1230 GOTO 740
1240 REM **AUSGABE**
1250 A$=DFE$(A-23,B-18)
1260 E=22
1270 F=4
1280 GOSUB 2510
1290 CALL HCHAR(17,A,32)
1300 CALL HCHAR(B,22,32)
1310 GOTO 760
1320 REM **ENDE**
1330 A$="WIRKLICH  ENDE ?"
1340 E=22
1350 F=4
1360 GOSUB 2510
1370 CALL KEY(0,K,S)
1380 IF S=0 THEN 1370
1390 IF (K=78)+(K=110)THEN 1
420
1400 IF (K=74)+(K=106)THEN 1
440
1410 GOTO 1370
1420 CALL HCHAR(22,4,32,16)
1430 GOTO 760
1440 CALL CLEAR
1450 END
1460 REM **ZEICHNEN**
1470 A=4
1480 B=4
1490 CALL GCHAR(B,A,T)
1500 IF T<>40 THEN 1670 ELSE
 1620
1510 CALL JOYST(1,X,Y)
1520 CALL KEY(1,K,S)
1530 IF K=18 THEN 1860
1540 IF X<>0 THEN 1570
1550 IF Y<>0 THEN 1720
1560 GOTO 1510
1570 IF (A+X/2<4)+(A+X/2>18)
THEN 2020
1580 GOSUB 1770
1590 A=A+X/2
1600 CALL GCHAR(B,A,T)
1610 IF T<>40 THEN 1670
1620 CALL HCHAR(B,A,43)
1630 CALL HCHAR(B,A+1,44)
1640 CALL HCHAR(B+1,A,45)
1650 CALL HCHAR(B+1,A+1,46)
1660 GOTO 1510
```

LISTINGS

```
1670 CALL HCHAR(B,A,90)
1680 CALL HCHAR(B,A+1,91)
1690 CALL HCHAR(B+1,A,88)
1700 CALL HCHAR(B+1,A+1,89)
1710 GOTO 1510
1720 IF (B-Y/2<4)+(B-Y/2>18)
THEN 2020
1730 GOSUB 1770
1740 B=B-Y/2
1750 CALL GCHAR(B,A,T)
1760 IF T<>40 THEN 1670 ELSE
 1620
1770 IF T<>40 THEN 1810
1780 CALL HCHAR(B,A,40,2)
1790 CALL HCHAR(B+1,A,40,2)
1800 RETURN
1810 CALL HCHAR(B,A,112)
1820 CALL HCHAR(B,A+1,113)
1830 CALL HCHAR(B+1,A,114)
1840 CALL HCHAR(B+1,A+1,115)
1850 RETURN
1860 IF T<>40 THEN 1930
1870 T=112
1880 IF A>10 THEN 1910
1890 Q(B/2-1,A/2-1)=1
1900 GOTO 1610
1910 W(B/2-1,A/2-5)=1
1920 GOTO 1610
1930 T=40
1940 GOSUB 1960
1950 GOTO 1610
1960 IF A>10 THEN 1990
1970 Q(B/2-1,A/2-1)=A/2
1980 RETURN
1990 W(B/2-1,A/2-5)=A/2-4
2000 RETURN
2010 REM **CHAR**
2020 EFD$=""
2030 GOSUB 1770
2040 FOR I=1 TO 8
2050 R=Q(I,1)*Q(I,2)*Q(I,3)*
Q(I,4)
2060 GOSUB 2130
2070 EFD$=EFD$&R$
2080 R=W(I,1)*W(I,2)*W(I,3)*
W(I,4)
2090 GOSUB 2130
2100 EFD$=EFD$&R$
2110 NEXT I
2120 GOTO 760
2130 IF R>30 THEN 2170
2140 IF R>15 THEN 2160
2150 ON R GOTO 2180,2200,222
0,2240,2260,2280,760,2300,76
0,2320,760,2340,760,760,2360
2160 ON R/2-9 GOTO 2380,760,
2400,760,760,2420
2170 ON R/20-1 GOTO 2440,246
0,760,760,2480
2180 R$="0"
2190 RETURN
2200 R$="8"
2210 RETURN
2220 R$="4"
2230 RETURN
2240 R$="2"
2250 RETURN
2260 R$="1"
2270 RETURN
2280 R$="C"
2290 RETURN
2300 R$="A"
2310 RETURN
2320 R$="9"
2330 RETURN
2340 R$="6"
2350 RETURN
2360 R$="5"
2370 RETURN
2380 R$="3"
2390 RETURN
2400 R$="E"
2410 RETURN
2420 R$="D"
2430 RETURN
2440 R$="B"
2450 RETURN
2460 R$="7"
2470 RETURN
2480 R$="F"
2490 RETURN
2500 GOTO 2500
2510 FOR I=1 TO LEN(A$)
2520 CALL HCHAR(E,(F+I)-1,AS
C(SEG$(A$,I,1)))
2530 NEXT I
2540 RETURN
2550 DATA 100,FF,101,0000000
0000000FF,102,010101010101010
01,103,8080808080808080
2560 DATA 104,0000000000000000
01,105,0000000000000008,106,0
1,107,8
2570 DATA 112,FF8080808080808
8,113,FF0101010101010101,114,8
0808080808080FF,115,0101010101
010101FF
2580 DATA 116,,40,FFFFFFFFFF
FFFFFF,43,FFFFFFFFFFFEFCF8,44,FFFFFFFFFF7F3F1F
2590 DATA 45,F8FCFEFFFFFFFFF
F,46,1F3F7FFFFFFFFFFF,90,FF80808080818387,91,FF0101010181C1E1
2600 DATA 88,87838180808080FF,89,E1C18101010101FF,38,0000183C3C18
2610 DATA 3,3,104,1,3,4,101,16,3,20,105,1,20,3,106,1,20,4,100,16,20,20,107,1
2620 DATA 12,21,104,1,12,22,101,2,12,24,105,1,13,21,102,
```

```
1,13,24,103,1,14,21,106,1
2630 DATA 14,22,100,2,14,24,
107,1,21,3,104,1,21,4,101,16
,21,20,105,1,22,20,103,1
2640 DATA 23,20,107,1,23,4,1
00,16,23,3,106,1,22,3,102,1,
18,23,104,1,18,24,101,4
2650 DATA 18,28,105,1,23,23,
106,1,23,24,100,4,23,28,107,
1
2660 DATA WAEHLEN,4,23,DEFIN
IEREN,6,23,AUSGABE,8,23,ENDE
,10,23,1234,16,24,A,19,21,B,
20,21,C,21,21,D,22,21
```

LISTINGS

Fortsetzung v. S. 45

CHAR

setzen oder einen neuen Character definieren, bzw. den vorherigen verbessern. Um das Zeichenfeld für eine neue Definition zu löschen, ist die 'C'-Taste zu drücken.

Ausgabe:
Falls man diesen Punkt aus dem Menü anwählt, muß zunächst, wie unter 'Wählen' beschrieben, ein Feld im 4x4 Raster angewählt werden. Auf Druck der Feuertaste hin wird der HEX-Code des betreffenden Characters unterhalb des Zeichenfeldes angezeigt. Selbstverständlich lassen sich Character im 4x4 Zeichenfeld durch Überschreiben auch wieder ändern.

Ende:
Wählt man diese Option, so erfolgt eine Sicherheitsabfrage, ob man wirklich aufhören will. Gibt man 'N' (=nein) ein, so kann man weitermachen, nach Eingabe von 'J' (=ja) ist das Programm beendet.

ALLE NEUNE

Mit diesem Programm kommt die Kegelbahn ins Haus.
Es können sich bis zu 10 Kegler beteiligen, jedoch ist auch der TI 99/4A gerne bereit, sein Können zu beweisen.
Damit bei Kegelabenden das Eingeben der Namen entfällt, können die Namen in das Vorprogramm einEDITiert werden.
Gekegelt wird mit Joystick 1. Die Wurfstärke erreicht man über die Dauer des Druckes des Aktionsknopfes.
Der TI 99/4A führt während des Kegelns die Gesamtholz, Pudel und Neunen jedes Keglers, um am Ende die Kegler nach dem Gesamtholz geordnet aufzulisten.

Programmhilfen:
Die Programmzeile 80 im Hauptprogramm dient zum Probelaufen, damit nicht jedes Mal das Vorprogramm nach einem gefundenen Tippfehler neu geladen werden muß.
Diese Zeile muß nach Fertigstellung des Hauptprogramms raus (REM vor oder ganz löschen), ansonsten sind es immer die zehn Computer, die kegeln.
Vor Eingabe der Zeilen 2000–2020 des Hauptprogramms, das Vorprogramm einmal durchlaufen lassen. Es erleichtert die Eingabe dieser drei Zeilen.

Bitte lesen Sie weiter auf Seite 50

```
10 ! ************************
11 ! *      ALLE NEUNE      *
12 ! *   (Vorprogramm)      *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *  Juergen Boehning    *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *    Cassettenrec.     *
24 ! *     Joystick 1       *
25 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     4871 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CLEAR :: CALL MAGNI
FY(2)
110 DIM SP$(10),SA$(10)
120 GOTO 140 :: K,S,ANZ,I=X
:: CALL VCHAR :: CALL HCHAR
:: CALL SPRITE :: CALL DELSP
RITE :: CALL COLOR
130 !@P-
140 CALL ASCI
150 CALL VCHAR(1,3,98,24)::
CALL VCHAR(1,32,96,24):: CAL
L HCHAR(1,3,97,30):: CALL HC
HAR(24,3,97,30)
160 FOR I=1 TO 14 :: CALL SP
RITE(#I,97,16,250,130,10,-10
,#I+14,97,16,250,130,10,10):
: FOR X=1 TO 50 :: NEXT X ::
 NEXT I
170 DISPLAY AT(3,7)SIZE(20):
"A L L E   N E U N E"
180 DISPLAY AT(6,5)SIZE(26):
"EIN KEGELSPIEL FUER 10"
190 DISPLAY AT(8,11):"KEGLER
" :: DISPLAY AT(10,5)SIZE(22
):"BRUEDER UND SCHWESTERN"
200 DISPLAY AT(16,2)SIZE(27)
:"(C)...JUERGEN BOEHNING"
210 DISPLAY AT(18,8)SIZE(20)
:"BORKENER STR. 1"
220 DISPLAY AT(20,8)SIZE(20)
:"2800 BREMEN 41"
230 DISPLAY AT(22,8)SIZE(20)
:"TEL.:0421/ 47 29 80"
240 CALL ASCII
250 DISPLAY AT(23,2)SIZE(27)
:"ANLEITUNG (J/N) "
260 CALL HALT(K)
270 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL MAGNIFY(1)
280 IF K=74 THEN CALL ANLEIT
UNG
290 X=0
300 CALL CLEAR
```

```
310 DISPLAY AT(1,1):"WIEVIEL
E KEGLER ? (1-10):"
320 ACCEPT AT(1,26)SIZE(3)VA
LIDATE(DIGIT,"-"):ANZ
330 IF ABS(ANZ)>10 OR ANZ=0
THEN 320
340 IF ANZ<0 THEN CALL AUTOM
ATIK(ANZ)
350 FOR I=1 TO ANZ
360 DISPLAY AT(2+I*2,1)SIZE(
12):I;TAB(5);"KEGLER :"
370 ACCEPT AT(3+I*2,2)VALIDA
TE(UALPHA)SIZE(-24):SP$(I)
380 IF SP$(I)="C" THEN X=X+1
 :: SP$(I)="COMP"&CHR$(X+64)
:: DISPLAY AT(3+I*2,2):SP$(I
)
390 NEXT I
400 CALL COLOR(1,1,1,2,1,1,3
,1,1,4,1,1)
410 CALL S(ANZ,SP$()):: CALL
 CLEAR
420 RUN "CS1"
430 !oder RUN"DSK1.PROGRAMM"
440 !************************
450 ! UNTERPROGRAMME
460 !************************
470 !
480 !@P+
490 SUB AUTOMATIK(ANZ)
500 GOTO 510 :: I,VERS=J ::
A$=""
510 ANZ=ABS(ANZ)
520 DATA RUDI RATLOS,HEINI G
LUECKLICH,EMIL HINKEBEIN,ANT
O LIEBLIG,BERTA SAUSEBRAUS,G
IDO HAMMERKOPF,HEINS SAGEMAL
,JOGI SCHREINICHTSO
530 !@P-
540 DATA FRED WISSMANN,NERO
DEMELACK
550 RESTORE 520
560 FOR I=1 TO ANZ
570 READ A$
580 DISPLAY AT(2+I*2,1):I;TA
B(5);"KEGLER :"
590 FOR VERS=1 TO 75 :: NEXT
 VERS
600 FOR J=1 TO LEN(A$)
610 DISPLAY AT(3+I*2,1+J):SE
G$(A$,J,1)
620 NEXT J
630 NEXT I
640 FOR VERS=1 TO 100 :: NEX
T VERS
650 !@P+
660 SUBEND
670 !
680 SUB ASCII
690 GOTO 700 :: A=I :: A$=""
700 DATA 100,000010284400000
710 !@P-
720 DATA 104,FFFFFFFFFFFFFF
F,105,FFFFFEFEFCFCF8F8,106,F
0F0E0E0C0C08080,107,FFFF7F7F
3F3F1F1F,108,0F0F07070303010
1
730 DATA 112,7F7FFFFFFFFFFF
F,113,3F3F3F7F7F7F7F,114,1
F1F1F1F1F1F3F3F,115,07070F0F
0F0F0F0F,116,0303030307070707
7,117,0101010101010303
740 DATA 118,FFFFFFFFFFFFFF
F,119,
750 DATA 120,FEFEFFFFFFFFFF
F,121,FCFCFCFCFEFEFEFE,122,F
8F8F8F8F8F8FCFC,123,E0E0F0F0
F0F0F0F0,124,C0C0C0C0E0E0E0E
0,125,8080808080C0C0
760 DATA 126,FFFFFFFFFFFFFF
F,127,
770 RESTORE 700
780 READ A,A$ :: CALL CHAR(A
,A$)
790 FOR I=1 TO 21
800 READ A,A$ :: CALL CHAR(A
,A$,A+16,A$)
810 NEXT I
820 !@P+
830 SUBEND
840 !
850 SUB ASCI
860 GOTO 870 :: A=I :: A$=""
 :: CALL CHAR
870 DATA 96,00007A7F7F7A0000
,97,183C183C3C3C3C00,98,0000
5EFEFE5E0000
880 !@P-
890 DATA 99,FFFFFFFFFFFFFF
900 RESTORE 870
910 FOR I=1 TO 4
920 READ A,A$ :: CALL CHAR(A
,A$)
930 NEXT I
940 !@P+
950 SUBEND
960 !
970 SUB ANLEITUNG
980 K=0
990 !@P-
1000 PRINT TAB(10);"ALLE NEU
NE": :"DIESES PROGRAMM STELL
T EIN": :"KEGELABEND MIT BIS
 ZU 10 ": :
1010 PRINT "KEGELSCHWESTERN
& -BRUEDERN": :"DAR.": :"DIE
 NAMEN WERDEN ZU BEGINN": :"
EINGEGEBEN.": :"WIRD DIE KEG
LER ANZAHL MIT": :
1020 PRINT "EINEN '-' DAVOR
EINGEGEBEN": :"SO GIBT DER C
OMPUTER NAMEN": :"VOR, DIE J
```

```
EDOCH NOCH": :"GEAENDERT WER
DEN KOENNEN."
1030 CALL HALT(K)
1040 PRINT "WIRD EIN 'C' EIN
GEGEBEN IST": :"DER COMPUTER
 BEREIT MIT ZU-": :"KEGELN."
: : :"NACHDEM DAS HAUPTPROGR
AMM": :"GELADEN IST,WIRD AUF
"
1050 PRINT :"DER LINKEN HAEL
FTE DIE": :"KEGELBAHN AUFGEB
AUT.": :"AUF DER RECHTEN HAE
LFTE OBEN": :
1060 PRINT "IST DER KEGLERNA
ME ZUFINDEN": :"DARUNTER IST
 DER WURFDISPLAY": :"MIT DEM
 GESAMT HOLZ.": :
1070 CALL HALT(K)
1080 PRINT "UNTER DEM SIND D
IE ANDEREN": :"KEGLER AUFGEL
ISTET MIT IHREM": :"GESAMT H
OLZ,PUDEL UND NEUN.": : : :
1090 PRINT "GEKEGELT WIRD IN
DEM MAN DIE": :"KUGEL MIT HI
LFE VON ": :"JOYSTICK 1 AN D
EN ABWURF-": :"PUNKT BRINGT
UND DANN DEN": :
1100 PRINT "AKTIONSKNOPF DRU
ECKT.": : :
1110 CALL HALT(K)
1120 PRINT "ES ERSCHEINT EIN
 WEISSER": :"PFEIL 'd' DER N
UN ZUR BE-": :"STIMMUNG DES
DRALLS DIENT.": :"NACHDEM DI
ESER EINGESTELLT": :
1130 PRINT "IST ERFOLGT DER
ABWURF.": :"DIE DAUER DES DR
UCKES AUF": :"DEN AKTIONSKNO
PF BESTIMMT": :"DIE WURFSTAE
RKE."
1140 CALL HALT(K)
1150 PRINT "WIRD ZULANGE GED
RUECKT": :"SPRINGT DIE KUGEL
 WIEDER ZU": :"IHRER AUSGANS
POSITION ZU-": :"RUECK UND M
USS NEU EINGE-": :"STELLT WE
RDEN.": :
1160 CALL HALT(K)
1170 PRINT "NACH JEDER RUNDE
 WIRD": :"GEFRAGT OB DIE NAE
CHSTE": :"RUNDE NOCH GEKEGEL
T WIRD.": :"ES IST EIN J(JA)
ODER EIN": :
1180 PRINT "N(NEIN) ZUDRUECK
EN.": : :
1190 CALL HALT(K)
1200 !@P+
1210 SUBEND
1220 !
1230 SUB HALT(K)
1240 GOTO 1260 :: S=0 :: CAL
L KEY
1250 !@P-
1260 DISPLAY AT(24,27):">>"
1270 CALL KEY(3,K,S):: IF S=
0 THEN 1270
1280 CALL CLEAR
1290 !@P+
1300 SUBEND
1310 !
1320 !
1330 !
1340 SUB S(A,S$())
1350 GOTO 1370 :: A$="" :: I
=J
1360 !@P-
1370 IF A<10 THEN A$="0"&STR
$(A)ELSE A$=STR$(A)
1380 CALL CHAR(64,A$)
1390 FOR I=1 TO A
1400 A$=""
1410 FOR J=1 TO LEN(S$(I))
1420 A$=A$&STR$(ASC(SEG$(S$(
I),J,1)))
1430 NEXT J
1440 CALL CHAR(30+I*3,A$)
1450 NEXT I
1460 !@P+
1470 SUBEND
```

LISTINGS

Fortsetzung von Seite 48

Alle Neune — Vorprogramm

K.S
CALL KEY Abfrage
I,X
Schleifenzähler
ANZ
die Anzahl der Kegler
SP$
für Keglernamen
SUB ASCI
Definiert die Zeichen für den Vorspann um
SUB ASCII
Definiert die Zeichen für das Hauptprogramm
SUB ANLEITUNG
Die genaue Bedienungsanleitung
SUB HALT(K)
Warte-Schleife für die Anleitung mit Rückmelde-Variable der gedrückten Taste
SUB S(A,SS())
sichert die eingegebenen Namen für das Hauptprogramm.

Alle Neune — Hauptprogramm

DU
Durchgangzähler
SP
Spielerzähler
WU
Wurfzähler
I
Schleifenzähler
A
Wurfstärke
D,B
Drall
W
Wertvariable
GES
Gesamt Zwischenspeicher
PU
Pudel Zwischenspeicher
NE
Neune Zwischenspeicher
GE ( )
Gesamtzähler
P ( )
Pudelzähler
N ( )
Neunenzähler
BW ( )
besten Wurf Speicher
ANZ
Anzahl der Spieler
SP$ ( )
Spieler Namen
SUB SETZT (Z,S,A,B,D)
Kugelsetzen, Drall geben
SUB ENDE (DU,SP$ ( ), GE ( ),...)
Rundenzähler, Endabfrage, wenn ja dann Bestentabelle

Bitte lesen Sie weiter auf Seite 54

LISTINGS

```
10 ! ************************
11 ! *      ALLE NEUNE      *
12 ! *  (Hauptprogramm)     *
13 ! *                      *
14 ! *     Copyright by     *
15 ! *                      *
16 ! *  Juergen Boehning    *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     7263 Bytes       *
28 ! *                      *
29 ! ************************
80 CALL CHAR(64,"10"):: FOR
I=1 TO 10 :: CALL CHAR(30+I*
3,"67797780"&STR$(64+I)&"32"
&RPT$(STR$(64+I),18)):: NEXT
 I
90 !*Zeile 80 ist fuer einen
*   Probelauf des Programmes
*       und muss nach
*   fertigstellung geloescht
*   werden.
100 CALL CLEAR :: RANDOMIZE
110 GOTO 130 :: A,B,D,W,I,SP
,GES,PU,NE=WU
120 CALL SPRITE :: CALL CHAR
 :: CALL KEY :: CALL JOYST
130 CALL DELS(ANZ,SP$())
140 DIM GE(10),P(10),N(10),B
W(10)
150 CALL MAGNIFY(1)
160 CALL CHARSET
170 DU=1
180 !@P-
190 CALL BILD(SP$(),ANZ)
200 FOR SP=1 TO ANZ
210 IF SP=1 THEN 230
220 FOR I=1 TO 300 :: NEXT I
230 DISPLAY AT(1,12)SIZE(16)
BEEP:SP$(SP):: DISPLAY AT(2,
16):"ccccccccccc cc"
240 FOR WU=1 TO 10
250 DISPLAY AT(2,4)SIZE(3):"
ava" :: DISPLAY AT(3,4)SIZE(
3):"ava"
260 CALL SPRITE(#1,97,16,21,
49,#2,97,16,13,49,#3,97,16,5
,49,#8,97,16,13,65,#9,97,16,
13,33)
270 !
280 IF SEG$(SP$(SP),1,4)="CO
MP" THEN CALL AUTO(A,B,D,W,S
P):: GOTO 310
290 CALL SETZT(185,49,A,B,D)
300 CALL WURF(A,D,B,W)
310 CALL WERTUNG(W,WU,GES,PU
,NE)
320 NEXT WU
330 GE(SP)=GE(SP)+GES :: P(S
P)=P(SP)+PU :: N(SP)=N(SP)+N
E :: PU,NE=0
340 BW(SP)=MAX(GES,BW(SP))
350 GES=0
360 CALL BEWERTUNG(GE(SP),P(
SP),N(SP),SP)
370 NEXT SP
380 CALL ENDE(DU,SP$(),GE(),
P(),N(),BW(),SP-1)
390 GOTO 200
400 !************************
410 ! UNTERPROGRAMME
420 !************************
430 !@P+
440 SUB SETZT(Z,S,A,B,D)
450 GOTO 470 :: K,Y,X1,Y1=ST
460 !@P-
470 CALL SPRITE(#11,91,9,Z,S
)
480 CALL KEY(1,K,ST):: IF ST
>0 THEN 530
490 CALL JOYST(1,X1,Y1)
500 S=S+X1/4 :: IF S<23 OR S
>77 THEN S=S-X1/4 :: GOTO 48
0
510 CALL LOCATE(#11,Z,S)
520 GOTO 480
530 Y=S
540 CALL SPRITE(#10,100,16,Z
,Y)
550 CALL KEY(1,K,ST):: IF ST
>0 THEN CALL DELSPRITE(#10):
: A=0 :: GOTO 600
560 CALL JOYST(1,X1,Y1)
570 Y=Y+X1/4 :: IF Y<S-5 OR
Y>S+5 THEN 550
580 CALL LOCATE(#10,Z,Y)
590 GOTO 550
600 D,B=Y-S
610 CALL KEY(1,K,ST):: IF ST
<0 THEN A=A+ST :: GOTO 610
620 IF A<-18 THEN S=49 :: GO
TO 470
630 CALL SOUND(-100,-3,0)
640 !@P+
650 SUBEND
660 !
670 SUB ENDE(DU,SP$(),GE(),P
(),N(),BW(),SP)
680 GOTO 700 K,S,X,GES,PU,NE
,B=I :: N$=""
690 !@P-
700 FOR I=1 TO 500 :: NEXT I
710 DISPLAY AT(1,10):"WIRD D
IE";DU+1;"RUNDE" :: DISPLAY
AT(2,10):"NOCH GEKEGELT ?"
720 FOR I=1 TO 150
730 CALL KEY(3,K,S):: IF S<>
```

```
0 THEN 750
740 NEXT I :: GOTO 760
750 IF K<>74 THEN 770
760 DU=DU+1 :: DISPLAY AT(1,
10):"" :: DISPLAY AT(2,10):"
" :: SUBEXIT
770 IF K<>78 THEN CALL SOUND
(-50,880,0):: GOTO 730
780 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL CHARSET ::
FOR I=1 TO 12 :: CALL COLOR(
1,2,1):: NEXT I
790 CALL CHAR(126,"011A244A5
2245880")
800 FOR I=1 TO SP-1 :: FOR X
=1+I TO SP :: IF GE(I)>GE(X)
THEN 830
810 N$=SP$(I):: GES=GE(I)::
PU=P(I):: NE=N(I):: B=BW(I):
: SP$(I)=SP$(X):: GE(I)=GE(X
):: P(I)=P(X):: N(I)=N(X)::
BW(I)=BW(X)
820 SP$(X)=N$ :: GE(X)=GES :
: P(X)=PU :: N(X)=NE :: BW(X
)=B
830 NEXT X :: NEXT I
840 DISPLAY AT(4,1):"NAME...
.GES..PU..NE..BW...~"
850 FOR I=1 TO SP :: DISPLAY
 AT(4+I*2,1):SEG$(SP$(I),1,6
);": ";TAB(9);STR$(GE(I));TA
B(13);P(I);TAB(17);N(I)
860 DISPLAY AT(4+I*2,22):STR
$(BW(I));TAB(26);STR$(INT(GE
(I)/DU+.5)/10):: NEXT I
870 CALL KEY(0,K,S):: IF S=0
 THEN 870 ELSE CALL CLEAR ::
 END
880 !@P+
890 SUBEND
900 !
910 SUB WURF(A,D,B,W)
920 GOTO 930 :: CALL PATTERN
 :: CALL MOTION :: Z=S
930 DEF PL(X)=53-INT((Z+4)/6
)
940 DEF PR(X)=44+INT((Z+4)/6
)
950 !@P-
960 CALL MOTION(#11,A,0)
970 CALL POSITION(#11,Z,S)
980 Z=INT(Z+A*1.6)
990 S=INT(S+SGN(49-S)+D)
1000 IF Z<4 OR Z>190 THEN CA
LL DELSPRITE(#11):: CALL WER
T(S,W):: SUBEXIT
1010 CALL SOUND(-500,110,40-
INT(Z/37.5+5)*3,-3,35-INT(Z/
37.5+5)*3)
1020 IF Z+A<47 THEN 1050
1030 CALL PATTERN(#11,96-INT
(Z/37.5+.5))
1040 IF S<PL(Z)OR S>PR(Z)THE
N CALL PUDEL(A,Z,S):: W=0 ::
 SUBEXIT
1050 CALL LOCATE(#11,Z,S)
1060 CALL SOUND(-500,110,40-
INT(Z/37.5+5)*3,-3,35-INT(Z/
37.5+5)*3)
1070 A=A-.1 :: D=D+(SGN(49-S
)+B)/4
1080 GOTO 960
1090 !@P+
1100 SUBEND
1110 !
1120 SUB BEWERTUNG(GES,PU,NE
,SP)
1130 !@P-
1140 DISPLAY AT(4+SP*2,18)SI
ZE(3):USING "###":GES
1150 DISPLAY AT(4+SP*2,23)SI
ZE(2):USING "##":PU
1160 DISPLAY AT(4+SP*2,27)SI
ZE(2):USING "##":NE
1170 !@P+
1180 SUBEND
1190 !
1200 SUB WERT(S,W)
1210 GOTO 1230 :: A,B,S,Q=I
1220 !@P-
1230 W=INT(ABS(49-S))
1240 IF W<3 THEN W=3 ELSE IF
 W>9 THEN W=9
1250 !@P+
1260 DATA 21,49,13,49,5,49,9
,41,2,4,9,57,2,6,17,41,3,4,1
7,57,3,6,13,65,13,33,3,4
1270 !@P-
1280 RESTORE 1260
1290 FOR I=1 TO W
1300 READ A,B
1310 IF S<>B THEN Q=SGN(B-S)
ELSE Q=1-INT(RND*2)*2
1320 IF I>3 AND I<8 THEN REA
D A,B :: DISPLAY AT(A,B)SIZE
(1):CHR$(97-Q):: GOTO 1340
1330 CALL SPRITE(#I,97-Q,16,
A,B+Q*4)
1340 NEXT I
1350 FOR I=1 TO 30 :: CALL S
OUND(-50,43000*-(W<>9)+990,I
,-43000*(W<>9)+1000,I,-43000
*(W<>9)+1010,I):: NEXT I
1360 !@P+
1370 SUBEND
1380 !
1390 SUB WERTUNG(W,WU,GES,PU
,NE)
1400 !@P-
1410 GES=GES+W :: PU=PU-(W=0
):: NE=NE-(W=9)
1420 DISPLAY AT(2,15+WU)SIZE
```

LISTINGS

```
(1):STR$(W):: DISPLAY AT(2,2
7)SIZE(2):USING "##":GES
1430 !@P+
1440 SUBEND
1450 !
1460 SUB PUDEL(A,Z,S)
1470 GOTO 1480 :: Y=X
1480 DEF PL(X)=53-INT((Z+4)/
6)
1490 DEF PR(X)=44+INT((Z+4)/
6)
1500 DEF LS(X)=INT(Z/37.5+5)
*3
1510 !@P-
1520 A=INT(A/6)*6
1530 IF S<49 THEN CALL LOCAT
E(#11,Z,PL(Z)-5)ELSE CALL LO
CATE(#11,Z,PR(Z)+5)
1540 CALL MOTION(#11,A,-SGN(
49-S)*A/6)
1550 CALL POSITION(#11,X,Y):
: IF X>50 THEN CALL PATTERN(
#11,96-INT(X/37.5+.5)):: CAL
L SOUND(-50,110,35-LS(X),-3,
40-LS(X)):: GOTO 1550
1560 IF S<49 THEN CALL LOCAT
E(#11,47,41)ELSE CALL LOCATE
(#11,47,57)
1570 CALL MOTION(#11,A,SGN(4
9-S)*A)
1580 CALL SOUND(-50,110,35-L
S(X),-3,40-LS(X))
1590 CALL POSITION(#11,X,Y):
: IF X>25 THEN 1590
1600 IF S<49 THEN CALL LOCAT
E(#11,20,20)ELSE CALL LOCATE
(#11,20,78)
1610 CALL MOTION(#11,A,-SGN(
49-S)*(A/3))
1620 CALL POSITION(#11,X,Y):
: IF X>4 AND X<190 THEN CALL
 SOUND(-50,110,35-LS(X),-3,4
0-LS(X)):: GOTO 1620
1630 CALL DELSPRITE(#11)
1640 !@P+
1650 SUBEND
1660 !
1670 SUB AUTO(A,B,D,W,SP)
1680 GOTO 1690 :: S,Z=I :: C
ALL SOUND :: CALL LOCATE ::
CALL DELSPRITE :: CALL POSIT
ION
1690 DIM SCH(10),DR(10),PA(1
0),G(10)
1700 !@P-
1710 CALL SPRITE(#11,91,9,18
5,49)
1720 IF G(SP)=0 OR G(SP)>3 T
HEN PA(SP)=INT(RND*52+25)::
SCH(SP)=-INT(RND*18+1):: DR(
SP)=SGN(49-S)*-(SCH(SP)<-6)
```

LISTINGS

```
1730 S=PA(SP):: A=SCH(SP)::
B,D=DR(SP)
1740 IF S=49 THEN 1780
1750 FOR I=49 TO S STEP SGN(
S-49)
1760 CALL LOCATE(#11,185,I)
1770 NEXT I
1780 CALL POSITION(#11,Z,S):
: CALL SPRITE(#10,100,16,Z,S
)
1790 IF D=0 THEN 1830
1800 FOR I=0 TO D STEP SGN(D
-S)
1810 CALL LOCATE(#10,Z,I+S)
1820 NEXT I
1830 FOR I=0 TO ABS(A)*10 ::
 NEXT I
1840 CALL DELSPRITE(#10)
1850 CALL SOUND(-100,-3,0)
1860 CALL WURF(A,D,B,W)
1870 IF W>8 THEN G(SP)=G(SP)
+1 ELSE G(SP)=0
1880 !@P+
1890 SUBEND
1900 !
1910 SUB BILD(SP$(),ANZ)
1920 GOTO 1930 :: CALL COLOR
 :: A,B=I :: A$=""
1930 CALL CLEAR :: CALL SCRE
EN(4)
1940 DATA 35,FF7F3F1F0F07030
1,36,FFFEFCF8F0E0C080,64,FFF
FFFFFFFFFFFFF,40,F8F8F8F0F0F
0E0E0,41,E0C0C0C0C0808080,42
,80C0E0F0F8FCFEFF
1950 !@P-
1960 DATA 43,0103070F1F3F7FF
F,45,0703030303010101,46,1F1
F1F0F0F0F0707,91,3C7EFFFFFFF
F7E3C,92,387CFEFEFE7C38,93,0
0183C7E7E3C18
1970 DATA 94,0000081C3E1C080
0,95,0000183C3C180000
1980 DATA 11,1,11,12,2,16,2,
16,2,1,2,1,2,1,2,1,16,12,15,
11,12,11,12,11,11,4,11,4
1990 !** A C H T U N G !  **
*     ALLE GROSSBUCHSTABEN
*  UND DAS ZEICHEN(<) IN DEN
*  ZEILEN  2000,2010,2020
*  MIT 'CTRL' DRUECKEN ! ! !
2000 DATA <pvvvvvxH,Aqvvvvvy
I,Brvvvvvz J,Csvvvvv{K,Dtvvvv
v!L,Euvvvvv}M,O<pvvvxH,OAqvv
vyI,OBrvvvzJ
2010 DATA OCsvvv{K,ODtvvv!L,
OEuvvv}M,OO<pvxH,OOAqvyI,OOB
rvzJ,OOCsv{K,OODtv!L,OOEuv}M
2020 DATA OO#*v+$,O#*vvv+$,#
*vvvvv+$,j)vvvvv-l,i(vvvvv.k
,h@@@@@@@h
```

```
2030 RESTORE 1940
2040 FOR I=1 TO 14
2050 READ A,A$ :: CALL CHAR(
A,A$)
2060 NEXT I
2070 FOR I=1 TO 14
2080 READ A,B :: CALL COLOR(
I,A,B)
2090 NEXT I
2100 FOR I=24 TO 1 STEP -1
2110 READ A$ :: DISPLAY AT(I
,1):A$
2120 NEXT I
2130 DISPLAY AT(2,16):"ccccc
ccccc cc" :: DISPLAY AT(4,16
):"  GES  PU  NE"
2140 FOR I=1 TO ANZ
2150 DISPLAY AT(4+I*2,11):SE
G$(SP$(I),1,5);TAB(18);"ccc.
.cc..cc"
2160 NEXT I
2170 !@P+
2180 SUBEND
2190 !
2200 SUB DELS(SP,S$())
2210 GOTO 2220 :: B$="" :: I
,J=A
2220 CALL CHARPAT(64,A$):: S
P=VAL(SEG$(A$,1,2))
2230 !@P-
2240 FOR I=1 TO SP
2250 CALL CHARPAT(30+I*3,A$)
2260 FOR J=1 TO 47 STEP 2
2270 A=VAL(SEG$(A$,J,2))
2280 IF A>31 THEN S$(I)=S$(I
)&CHR$(A)ELSE 2320
2290 IF J=15 THEN CALL CHARP
AT(31+I*3,B$):: A$=A$&B$
2300 IF J=23 THEN CALL CHARP
AT(32+I*3,B$):: A$=A$&B$
2310 NEXT J
2320 NEXT I
2330 !@P+
2340 SUBEND
```

Fortsetzung von Seite 50

ALLE NEUNE

SUB BEWERTUNG (GES,PU,NE,SP)
Trägt die Werte in die Liste ein
SUB WERT (S,W)
errechnet den Wert und wirft die Kegel um
SUB WERTUNG (W,WU,GES,PU,NE)
Trägt die Werte in das Display ein
SUB PUDEL(A,Z,S)
Lenkt die Kugel in der Pudelbahn, wenn diese dort hineingefallen ist
SUB AUTO(A,B,D,W,SP)
Der Computer wirft
SUB BILD(SP$( ),ANZ)
Definiert letzte Zeichen, setzt Farbe und erstellt die Bildschirmmaske
SUB DELS(SP,S$ ( ))
Liest die im Vorprogramm eingegebenen Namen wieder aus.

LISTINGS

TI-PITFALL

Spielregeln:

Das vorliegende Spiel ist dem Original-Pitfall nachempfunden. Ziel ist es, durch das Urwaldlabyrinth zu laufen um Schätze zu finden. Auf dem Weg lauern zahlreiche Gefahren auf den Helden, die er überspringen oder auf andere Weise überwinden kann. Dies sind die groben Grundregeln des Spieles. In mancher Hinsicht unterscheidet sich das Programm jedoch von dem Original. So besitzt das Spiel z.B. 90 Bilder. In diesen Bildern existieren folgende Objekte:
Fässer
Schlangen
See (Variaton: mit Schildkröten oder verschwindender See)
Vögel
Schätze
Leitern
Mauern
Löcher
Skorpione (unterirdisch)
und natürlich die Spielfigur.

Besonderheitden der Objekte:

a) Die Fässer existieren in einigen Bildern. Sie treten einzeln oder zu zweit auf und rollen durch den Bildschirm. Die Figur muß die Fässer überspringen, um keinen Punkteverlust zu erhalten.
b) Die Schlange muß ebenfalls übersprungen werden. Sie bewegt sich nicht. Eine Berührung kostet den Helden ein Bildschirmleben.
c) Eine andere Bildform ist der See. Eine Variation davon ist der See mit Schildkröten. Um ihn zu überwinden, muß die Figur auf jede der 3 Schildkröten springen. Dieser Vorgang sollte zügig vollzogen werden, da die Schildkröten die Neigung haben, sich nach einiger Zeit umzufärben. Befindet sich der Held dann auf der Schildkröte, verliert er ein Leben.
Um den See ohne Schildkröten zu überwinden, existiert in den entsprechenden Bildern ein Vogel, der durch die Szenerie fliegt. Dieser kann von dem Helden angesprungen werden. Gelingt diese Aktion, wird die Figur sicher ins nächste Bild getragen. Existiert an einem See kein Vogel, so ist dies ein verschwindender See.
Die Figur muß warten, bis der See verschwindet und dann ans andere Ufer laufen. Wie bei allen anderen See-Bildern führt auch hier jeglicher Sturz ins Wasser zum Tod.
d) Die Leitern können benutzt werden, um den unterirdischen Tunnel zu erreichen, oder um nach einem Sturz in die Tiefe erneut hochzuklettern. Fällt man die Öffnung, in der sich die Leiter befindet einfach herunter, verliert man Punkte. Dasselbe gilt für den Sturz durch ein Loch.
Allgemein kann gesagt werden, daß der unterirdische Tunnel eine Alternative zu den oberen Bildern darstellt. Hindernisse dieses Tunnels sind der Skorpion, den man überspringen muß, um nicht ein Leben zu verlieren, und die Mauern, die nicht überwunden werden können. Die Figur muß, falls sie auf eine Mauer trifft, den eingeschlagenen Weg zurückgehen.
e) Und nun zur Spielfigur selbst. Das weiße Männchen hat verschiedene Funktionen. Es kann sich nach rechts und links bewegen, springen und Leitern hoch oder runter klettern. Zu Beginn des Spiels hat man 3 Bildschirmleben. Bleibt die Figur länger als ungefähr 6 Sekunden auf einer Stelle stehen, so verliert sie ein Leben.

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *     TI-PITFALL       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *    Achim Truger      *
17 ! *  Michael Buerger     *
18 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *    Cassettenrec.     *
23 ! *(oder Disk+32K-Erw.)  *
24 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    10212 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CLEAR :: DISPLAY AT
(1,4):"**** TI-PITFALL ****"
:"...====================" :
: DISPLAY AT(12,8):"BITTE WA
RTEN" :: OPTION BASE 1 :: DI
M BI(90,4)
110 RESTORE :: FOR A=1 TO 90
 :: FOR A1=1 TO 4 :: READ BI
(A,A1):: NEXT A1 :: NEXT A
120 CALL CLEAR :: CALL SCREE
N(13):: CALL COLOR(1,2,13,2,
2,2,3,16,13,4,16,13,5,4,4,6,
6,6,7,7,16,10,16,13,10,11,2)
130 CALL COLOR(11,12,2):: CA
LL MAGNIFY(3)
140 CALL CHAR(40,"FFFFFFFFFF
FFFFFF",44,"0107030302030708
1323030202041810806E0C0C004C8
F0C0C0C0C0402010101C")
150 CALL CHAR(58,"1010101010
101010",60,"0107030320130F03
0303030204080838806E0C0C040C0
E0D0C8C4C04040201808")
160 CALL CHAR(64,"FFFFFFFFFF
FFFFFF",68,"0000000201000000
0000010300030003000000C0E0D0
1818204080E000E404F8")
170 CALL CHAR(72;"FFFFFFFFFF
FFFFFF",76,"0000000006010103
070E0E0E0F0E0F07000000003040
C0607038F838B83878F0")
180 CALL CHAR(80,"0000000000
0000000000000073D77F1F3F000000
0000000000000000F0F8FCFEFF")
190 CALL CHAR(84,"F7F7F700EF
EFEF",88,"0000000000000000001
3108340307797300000000000000
F8C4C28242BAFCF804")
200 CALL CHAR(92,"0107030302
0303030303030202020380E0C0
C000C0D0E0C0C0C04040407080")
210 CALL CHAR(96,"0107030300
030B07030303020202 0E0180E0C0
C040C0C0C0C0C0C040404040C0")
220 CALL CHAR(100,"0107030302
03070B1323033F2020000080E0C
0C004C8F0C0C0C2C2FE00000000"
)
230 CALL CHAR(104,"0107030320
130F030343437F0000000080E0C
0C040C0E0D0C8C4C0FC04040000"
)
240 CALL CHAR(109,"FFFFFFFFF
FFFFFFF",112,"FFFFFFFFFFFFFF
FF",113,"0F0F00000F0F",114,"
F0F00000F0F0")
250 CALL CHAR(116,"000000000
0000000103070F182230140F00000
0000000F8FCFEFEFCF830A04080"
)
260 CALL CHAR(120,"000000000
000000001070323110B070300000
0000000000080E0C0C488D0E0C0"
)
270 CALL CHAR(124,"000000000
000000001070308090B070300000
0000000000080E0C0D090D0E0C0"
)
280 CALL CHAR(128,"000000000
0E1F1783C9E7F3F1F0103050000000
000E0F0D8FC7820F0F0E0008040"
)
290 CALL CHAR(132,"000000000
000001F236361627B3F1F2000000
0000000000000808C102CC0E09ECE"
)
300 CALL CHAR(136,"000000000
70F1B3F1E040F0F0700010200000
00000878F1E3C79FEFCF880C0A0"
)
310 CALL HCHAR(24,1,109,32):
: CALL HCHAR(18,1,40,192)::
CALL HCHAR(16,1,109,64):: CA
LL HCHAR(13,1,112,96):: CALL
 HCHAR(8,1,64,160)
320 FOR A=7 TO 27 STEP 10 ::
 CALL VCHAR(8,A,40,5):: NEXT
 A
330 CALL HCHAR(7,1,64,5):: C
ALL HCHAR(7,9,64,7):: CALL H
CHAR(7,19,64,7):: CALL HCHAR
(7,29,64,4)
340 CALL HCHAR(6,1,64,4):: C
ALL HCHAR(6,10,64,5):: CALL
HCHAR(6,20,64,5):: CALL HCHA
R(6,30,64,3)
350 C=0 :: BE=1 :: B,U=0 ::
M=10 :: P=500 :: DISPLAY AT(
1,24):P :: E=2 :: CALL HCHAR
(1,2,58,E):: GOSUB 690
360 GOSUB 490 :: GOSUB 990 :
: GOSUB 1110 :: CALL KEY(0,T
,S):: IF T=47 OR T=80 OR T=5
```

```
1 OR T=32 THEN Z=0 :: GOTO 3
90 ELSE CALL MOTION(#1,0,0)
370 IF M=-10 THEN CALL PATTE
RN(#1,96)ELSE CALL PATTERN(#
1,92)
380 Z=Z+1 :: IF Z>20 THEN CA
LL MOTION(#1,0,0):: Z=0 :: G
OSUB 1040 :: GOTO 360 ELSE 3
60
390 IF T=47 THEN CALL PATTER
N(#1,44):: M=10 :: CALL MOTI
ON(#1,0,M):: GOTO 360
400 IF T=80 THEN CALL PATTER
N(#1,60):: M=-10 :: CALL MOT
ION(#1,0,M):: GOTO 360
410 IF T<>51 THEN 450 ELSE I
F U=0 OR BI(B,1)=3 OR X<115
OR X>130 THEN 450
420 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,169,120):: CALL
 MOTION(#1,-10,0):: CALL SOU
ND(1000,-4,0):: FOR A=1 TO 2
15 :: NEXT A :: CALL MOTION(
#1,0,0)
430 CALL KEY(0,T,S):: IF T<>
47 AND T<>80 AND T<>69 THEN
430 ELSE IF T=69 THEN GOSUB
580 :: GOTO 360 ELSE IF T=47
 THEN CALL LOCATE(#1,95,135)
ELSE CALL LOCATE(#1,95,105)
440 U=0 :: GOTO 360
450 IF T<>32 THEN 360 ELSE C
ALL MOTION(#1,0,0):: GOSUB 9
90 :: CALL POSITION(#1,Y,X):
: IF M=-10 THEN CALL PATTERN
(#1,104)ELSE CALL PATTERN(#1
,100)
460 CALL LOCATE(#1,Y-8,X)::
CALL MOTION(#1,0,M):: BE=.5
:: FOR V=1 TO 5 :: GOSUB 490
 :: GOSUB 990 :: GOSUB 1110
:: IF BW=1 THEN BE=1 :: GOTO
 360
470 NEXT V :: CALL MOTION(#1
,0,0):: BE=1
480 IF M=-10 THEN CALL LOCAT
E(#1,Y+8,X-7):: CALL PATTERN
(#1,60):: GOTO 360 ELSE CALL
 LOCATE(#1,Y+8,X+7):: CALL P
ATTERN(#1,44):: GOTO 360
490 CALL POSITION(#1,Y,X)::
IF X>240 OR X<14 THEN 660 EL
SE BW=0
500 IF U=0 THEN 570
510 IF BI(B,2)=0 THEN RETURN
520 IF BI(B,2)=2 THEN 540 EL
SE IF X>38 OR X<18 THEN BW=0
 :: RETURN ELSE BW=1 :: CALL
 MOTION(#1,0,0):: CALL SOUND
(200,-7,0,200,5)
530 IF M=10 THEN CALL LOCATE
(#1,169,17):: CALL PATTERN(#
1,92):: RETURN ELSE CALL LOC
ATE(#1,169,39):: CALL PATTER
N(#1,96):: RETURN
540 IF X>230 OR X<212 THEN B
W=0 :: RETURN ELSE BW=1 :: C
ALL MOTION(#1,0,0):: CALL SO
UND(200,-7,0,200,5)
550 IF M=10 THEN CALL LOCATE
(#1,169,210):: CALL PATTERN(
#1,92)ELSE CALL LOCATE(#1,16
9,231):: CALL PATTERN(#1,96)
560 RETURN
570 IF BI(B,1)=3 OR BI(B,1)=
4 THEN GOSUB 1140 :: RETURN
ELSE IF T=32 AND Y<90 THEN R
ETURN ELSE IF X<110 OR X>130
 THEN 600 ELSE CALL KEY(0,T,
S)
580 IF T<>69 THEN 640 ELSE C
ALL MOTION(#1,0,0):: CALL LO
CATE(#1,120,120):: CALL SOUN
D(1000,-4,0):: CALL MOTION(#
1,10,0)
590 FOR A=1 TO 230 :: NEXT A
 :: CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,169,120):: U=1
:: RETURN
600 IF BI(B,1)=1 THEN RETURN
610 IF X<38 OR X>63 THEN IF
X<182 OR X>204 THEN RETURN
620 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,137,X):: CALL M
OTION(#1,10,0):: FOR A=200 T
O 110 STEP -5 :: CALL SOUND(
-100,A,0):: NEXT A
630 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,169,X):: U=1 ::
 P=P-100 :: GOSUB 1260 :: DI
SPLAY AT(1,24):P :: RETURN
640 CALL LOCATE(#1,120,120):
: CALL MOTION(#1,10,0):: FOR
 A=240 TO 110 STEP -5 :: CAL
L SOUND(-100,A,0):: NEXT A :
: CALL MOTION(#1,0,0):: U=1
:: CALL LOCATE(#1,169,120)
650 P=P-100 :: GOSUB 1260 ::
 DISPLAY AT(1,24):P :: RETUR
N
660 CALL DELSPRITE(ALL):: IF
 X>240 THEN 690
670 B=B-1 :: R=0 :: IF B<1 T
HEN B=90
680 GOTO 700
690 B=B+1 :: R=1 :: IF B>90
THEN B=1
700 ON BI(B,1)GOSUB 830,840,
860,870 :: ON BI(B,2)+1 GOTO
 710,720,730
710 CALL VCHAR(18,5,40,6)::
CALL VCHAR(18,29,40,6):: CAL
```

```
L SPRITE(#10,88,16,169,130,0
,-M*.4):: IF M=-10 THEN CALL
 PATTERN(#10,132):: GOTO 740
 ELSE 740
720 CALL VCHAR(18,29,40,6)::
 CALL VCHAR(18,5,84,6):: GOT
O 740
730 CALL VCHAR(18,5,40,6)::
CALL VCHAR(18,29,84,6)
740 IF BI(B,4)=0 THEN 770
750 FLM=.8*M :: IF BI(B,4)=1
 THEN CALL SPRITE(#2,76,7,98
,230):: GOTO 770 ELSE IF BI(
B,4)=2 THEN CALL SPRITE(#2,6
8,5,100,220):: GOTO 770
760 IF M=10 THEN CALL SPRITE
(#6,128,15,73,1,0,FLM)ELSE C
ALL SPRITE(#6,136,15,73,240,
0,FLM)
770 IF BI(B,1)=3 AND BI(B,3)
=0 THEN CALL SPRITE(#3,80,2,
97,90,#4,80,2,97,122,#5,80,2
,97,153)
780 IF BI(B,3)>2 THEN 800
790 FOR A=1 TO BI(B,3):: CAL
L SPRITE(#A+2,116,7,98,250,0
,-10):: FOR A1=1 TO 110 :: N
EXT A1 :: NEXT A
800 IF U THEN YP=169 ELSE YP
=95
810 IF R THEN CALL SPRITE(#1
,92,16,YP,14)ELSE CALL SPRIT
E(#1,96,16,YP,240)
820 BW=1 :: RETURN
830 GOSUB 980 :: CALL HCHAR(
14,16,40,2):: CALL VCHAR(16,
16,113,8):: CALL VCHAR(16,17
,114,8):: RETURN
840 GOSUB 980 :: CALL HCHAR(
14,7,40,2):: CALL HCHAR(14,1
6,40,2):: CALL HCHAR(14,25,4
0,2)
850 CALL VCHAR(16,16,113,8):
: CALL VCHAR(16,17,114,8)::
RETURN
860 GOSUB 980 :: CALL HCHAR(
14,10,72,14):: WA=1 :: RETUR
N
870 GOSUB 980 :: WA=0 :: F=8
 :: RETURN
880 DATA 1,1,1,0,2,0,2,0,3,1
,0,0,2,0,0,2,4,0,0,0,1,2,0,1
,4,0,0,3,3,0,0,3,2,0,2,0
890 DATA 4,0,2,0,1,1,0,1,2,0
,0,2,1,0,1,0,3,0,0,0,1,2,0,1
,3,0,0,0,2,2,2,0,3,0,3,3
900 DATA 4,0,0,2,2,1,1,0,1,0
,2,0,4,0,0,3,4,2,0,1,2,1,2,0
,3,0,0,0,1,0,1,0,3,0,0,3
910 DATA 2,2,0,2,4,1,1,0,3,0
,2,3,2,0,2,0,1,1,2,0,2,0,0,1
,1,2,0,1,3,0,0,0,4,0,0,2
920 DATA 2,0,1,0,1,0,2,0,2,0
,1,0,4,0,0,3,1,2,0,0,4,0,1,3
,3,0,2,3,3,0,0,0,1,0,0,2
930 DATA 4,0,0,0,1,0,2,0,2,2
,1,0,3,1,0,3,3,2,0,0,2,1,0,1
,1,1,0,2,3,0,2,3,4,0,1,3
940 DATA 2,1,2,0,4,0,0,3,2,0
,0,2,2,2,0,1,3,0,0,3,4,0,0,2
,3,0,3,3,1,0,2,0,2,0,0,2
950 DATA 4,0,2,0,3,0,0,0,1,2
,1,0,3,1,0,3,4,0,0,2,2,2,1,0
,3,0,1,3,2,2,2,0,1,1,0,2
960 DATA 1,0,0,0,3,0,0,0,4,0
,0,1,1,1,0,0,3,0,0,3,3,0,3,3
,4,0,0,3,2,2,2,0,1,0,0,2
970 DATA 1,0,2,0,3,1,0,0,4,0
,0,2,1,0,0,0,1,0,0,0,3,0,0,0
,3,0,0,0,4,1,0,1,2,0,2,0
980 CALL HCHAR(14,7,112,20):
: CALL HCHAR(16,1,109,64)::
CALL VCHAR(18,16,40,6):: CAL
L VCHAR(18,17,40,6):: RETURN
990 IF SC THEN RETURN
1000 CALL COINC(ALL,HIT):: I
F HIT=0 THEN RETURN ELSE CAL
L MOTION(#1,0,0):: IF U THEN
 1040
1010 IF Y<90 THEN 1080 ELSE
ON BI(B,4)+1 GOTO 1020,1030,
1040,1020
1020 CALL SOUND(50,-6,0):: P
=P-20 :: GOSUB 1260 :: DISPL
AY AT(1,24):P :: RETURN
1030 FOR A=0 TO 15 :: CALL S
OUND(-100,2000,A):: NEXT A :
: P=P+500 :: DISPLAY AT(1,24
):P :: CALL DELSPRITE(#2)::
BI(B,4)=0 :: C=C+1 :: IF C>9
 THEN 1290 :: RETURN
1040 GOSUB 1270 :: E=E-1 ::
CALL HCHAR(1,2,32,2):: IF E=
-1 THEN 1220 ELSE CALL HCHAR
(1,2,58,E)
1050 BW=1 :: FOR A=1 TO 200
:: NEXT A :: CALL DELSPRITE(
#1):: IF U THEN 1070
1060 CALL SPRITE(#1,44,16,40
,15):: FOR A=40 TO 95 :: CAL
L LOCATE(#1,A,15):: NEXT A :
: RETURN
1070 CALL SPRITE(#1,44,16,13
7,15):: FOR A=137 TO 169 ::
CALL LOCATE(#1,A,15):: NEXT
A :: RETURN
1080 BE=.4 :: CALL MOTION(#1
,0,FLM):: CALL SOUND(100,300
,0):: CALL SOUND(400,400,0)
1090 GOSUB 1110 :: CALL POSI
TION(#1,Y,X):: IF X>240 OR X
<12 THEN 1100 ELSE CALL KEY(
```

```
0,T,S):: IF T<>69 THEN 1090
1100 CALL MOTION(#1,0,0):: C
ALL SOUND(100,400,0):: CALL
SOUND(400,300,0):: Z=0 :: CA
LL LOCATE(#1,95,X):: BW,BE=1
 :: RETURN
1110 IF BI(B,1)<>4 THEN RETU
RN ELSE IF F<14 THEN F=F+BE
:: RETURN
1120 CALL GCHAR(14,10,ZE)::
IF ZE=72 THEN CALL HCHAR(14,
10,112,14):: WA=0 ELSE CALL
HCHAR(14,10,72,14):: WA=1
1130 F=0 :: RETURN
1140 IF Y<90 OR WA=0 OR X<62
 OR X>180 THEN SC=0 :: KA=0
:: RETURN
1150 IF BI(B,1)=4 THEN 1190
1160 CALL COINC(#1,#3,8,SC):
: IF SC=0 THEN CALL COINC(#1
,#4,8,SC):: IF SC=0 THEN CAL
L COINC(#1,#5,8,SC)
1170 IF SC=0 THEN 1190 ELSE
KA=KA+1 :: IF KA<2 THEN RETU
RN ELSE CALL MOTION(#1,0,0):
: CALL COLOR(#3,12,#4,12,#5,
12):: KA=0
1180 FOR A=1 TO 20 :: NEXT A
 :: CALL COLOR(#3,2,#4,2,#5,
2):: GOTO 1040
1190 CALL MOTION(#1,0,0):: I
F X<90 THEN X=76 ELSE IF X<1
22 THEN X=105 ELSE IF X<153
THEN X=138 ELSE X=165
1200 CALL LOCATE(#1,Y,X)
1210 FOR A=1 TO 10 :: CALL P
ATTERN(#1,120):: FOR A1=1 TO
 50 :: NEXT A1 :: CALL PATTE
RN(#1,124):: FOR A1=1 TO 50
:: NEXT A1 :: NEXT A :: GOTO
 1040
1220 CALL KEY(0,T,S):: IF S=
0 THEN 1220 ELSE CALL DELSPR
ITE(ALL):: CALL CLEAR :: CAL
L CHARSET :: DISPLAY AT(12,1
):"WOLLEN SIE NOCHMAL(J/N)?"
1230 IF C>9 THEN DISPLAY AT(
5,9):"GESCHAFFT!!!"
1240 DISPLAY AT(20,12):"> <"
1250 CALL KEY(0,T,S):: IF S=
0 THEN 1250 ELSE CALL HCHAR(
20,15,T):: IF T=74 THEN 100
ELSE IF T=78 THEN END ELSE 1
250
1260 IF P<1 THEN DISPLAY AT(
1,24):"0000" :: CALL HCHAR(1
,2,32,3):: GOTO 1220 ELSE RE
TURN
1270 FOR A=1 TO 3 :: CALL SO
UND(150,165,0):: CALL SOUND(
10,110,30):: NEXT A :: CALL
```

LISTINGS

```
SOUND(600,131,0):: CALL SOUN
D(100,110,30)
1280 FOR A=1 TO 3 :: CALL SO
UND(150,147,0):: CALL SOUND(
10,110,30):: NEXT A :: CALL
SOUND(600,123,0):: RETURN
1290 CALL SOUND(200,500,0)::
 CALL SOUND(600,600,0):: CAL
L HCHAR(3,1,33,32):: GOTO 12
20
```

FLUG-SIMULATOR

Vor dem Laden CALL FILES(1) und NEW eingeben.
Das Programm ist in Modulen aufgebaut, um eine gewisse Übersichtlichkeit zu wahren. Jedoch ist einschränkend zu sagen, daß trotzdem diverse Sprünge vorhanden sind.
In Zeile 100 und 110 werden die Unterprogramme Einleitung und Start aufgerufen. Hier wird auch auf die wichtigsten Daten des Flugzeugs hingewiesen.
Es wird nun das Unterprogramm "zeidef" (Zeichendefinitionen) aufgerufen: Grenze, Berge, Raketen, Flakstationen, Ziele, Neigungsanzeige und natürlich das Flugzeug werden definiert. Im Unterprogramm "bild" wird – mit Hilfe des Zufallsgenerators – die Karte aufgebaut.
Es dürfen sich dabei keine Sprites überlappen, damit die spätere Abfrage schneller, genauer und einfacher funktioniert.
Das in Zeile 150 aufgerufene Unterprogramm "display" für Text wird später nicht mehr aufgerufen, um Zeit zu sparen. Infolgedessen arbeitet "steuerdatendisplay", das während des Programms etwa alle 1.5 Sekunden aufgerufen wird, mit Ausgabeformatierung, um den Text nicht anzugreifen.
Die Zeilen 170, 180 und 190 sind nun wichtiger: In Zeile 170 wird der Teil aufgerufen, der feststellt, ob und welche Funktionen der Pilot angesteuert hat. Von hier aus verzweigt sich das Programm weiter in die Berechnungen (z.B. Tank, Neigungswinkel, Kurs, Steigen und Sinken, Höhe, Mach usw.) Außerdem werden die wichtigsten Kontrollen angesteuert, die sich hauptsächlich im Unterprogramm "universalkontrolle" befinden. Bei gegebenem Anlaß verzweigt sich das Programm von hier aus noch weiter (Landungen o.ä.).
Im Teil Bewegung wird der Kurs (im Gradmaß) in Bewegung des stilisierten Flugzeugs auf der Karte umgerechnet.
Nach Abschluß des Fluges erscheint noch ein knapper Kommentar zum Flug, der die Leistungen bzw. Fehlschläge des Piloten berücksichtigt.
Je nach Joystick(-Interface) bzw. Gewohnheit kann in 1520 und 1530 variiert werden. Das Flugzeug sollte steigen, wenn man am Joystick zieht. Sollte das Flugzeug auf die falsche Seite fliegen, so sind die Parameter "vv" und "seiqu" (bedeutet Seite/Quer) zu vertauschen.
In 700 kann die Länge der Schleife verändert werden, so daß je nach Können mehr oder weniger Berge auf der Karte sind.
In 2710 kann der Höhengrenzwert verändert werden, so daß man vom Radar in unterschiedli-

Bitte lesen Sie weiter auf Seite 64

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *   FLUGSIMULATOR      *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Karsten Knobel     *
17 ! *                      *
19 ! * Benoetigte Geraete.* *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! *    Cassettenrec.     *
24 ! *(oder Disk+32K-Erw.)* *
25 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    11768 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL EINLEITUNG
110 CALL START
120 GOSUB 210
130 GOSUB 630
140 GOSUB 400
150 GOSUB 1190
160 GOSUB 1350
170 GOSUB 1520
180 GOSUB 2620
190 GOSUB 2700
200 GOTO 160
210 !
220 DISPLAY AT(12,7)ERASE AL
L:"MOMENT PLEASE!"
230 CALL CHAR(91,"0000001000
000000",92,"8100210088004100
0000000000000000000000000000
0000000000000000000")
240 CALL CHAR(97,"3C42BDBDBD
BD423C")
250 CALL CHAR(98,"8181818181
818181",99,"FF000000000000FF
")
260 CALL CHAR(100,"01071F1F3
F3F7F3F3F7F7F3FBF1A0500BCFFF
FFFFFFFFFFEFCFCFEFFFFFFFF66"
)
270 CALL CHAR(104,"000040000
8000000000100000000001000400
0020000100000000200000001000"
)
280 CALL CHAR(136,"C0C000000
0000000000000000000000000000
0000000000000000000000000000"
)
290 CALL CHAR(108,"000800080
0080008")
300 DATA 00000000FFFFFFFF,00
00000FFFFFFFFF,0000030F3FFFF
FFF,0103070F1F3F7FFF,03070F0
F1F1F3F3F,0F0F0F0F1F1F1F1F,0
F0F0F0F0F0F0F0F
310 DATA 1F1F1F1F0F0F0F0F,3F
3F1F1F0F0F0707,FF7F3F1F0F070
301,FFFF3F0F03000000,FFFFFF0
F00000000,FFFFFFFF00000000,F
FFFFFF000000000
320 DATA FFFFFFFCF0C00000,FF
FEFCF8F0E0C080,F8F8F0F0E0E0C
0C0,F8F8F8F8F0F0F0F0,F0F0F0F
0F0F0F0F0,E0E0E0E0F0F0F0F0,C
0C0E0E0F0F0F8F8
330 DATA 80C0E0F0F8FCFEFF,00
00C0F0FCFFFFFF,000000F0FFFFF
FFF
340 FOR DEFI=109 TO 132
350 READ DEFI$
360 CALL CHAR(DEFI,DEFI$)
370 NEXT DEFI
380 CALL CLEAR :: CALL COLOR
(8,13,1)
390 RETURN
400 !
410 NIG=0
420 K=0
430 NEG=0
440 HOEHE=0
450 VVERT=0
460 NEIG=109
470 NBRE$="OFF"
480 TANK=100
490 KOMP=0
500 MACH=0
510 D1=0
520 D2=0
530 SCHUB=0
540 FWRK$="OUT"
550 HOEWI=1
560 DUEPPEL=6
570 FEHLER=0
580 FLUG=0
590 SP=5
600 FPS$="IN"
610 MACHMIN=0.6 :: MACHMAX=0
.8
620 RETURN
630 !
640 CALL DELSPRITE(ALL)
650 CALL CLEAR
660 DISPLAY AT(21,1)BEEP:"BI
TTE RAUCHEN EINSTELLEN.....
...UND ANSCHNALLEN !"
670 FOR ZEILE=2 TO 18
680 CALL HCHAR(ZEILE,1,91,25
)
690 NEXT ZEILE
700 FOR S=8 TO 21
710 CALL MAGNIFY(3)
720 GOTO 750
730 RANDOMIZE :: Y=INT(RND*1
20+8):: RANDOMIZE :: X=INT(R
ND*70+84)
740 RETURN
```

```
750 GOSUB 730
760 CALL SPRITE(#S,100,7,Y,X
)
770 FOR T=1 TO 10 :: NEXT T
780 CALL COINC(ALL,TR):: IF
TR=-1 THEN 790 ELSE 800
790 GOSUB 730 :: CALL LOCATE
(#S,Y,X):: GOTO 770
800 NEXT S
810 CALL COINC(ALL,TR):: IF
TR=-1 THEN 630
820 CALL VCHAR(2,26,108,18)
830 CALL HCHAR(6,28,99,2)
840 CALL HCHAR(15,29,99,2)
850 CALL VCHAR(11,28,98,2)
860 FOR TARN=2 TO 16 STEP 2
870 CALL VCHAR(TARN,8,104)
880 CALL VCHAR(TARN+1,8,105)
890 CALL HCHAR(TARN,9,106)
900 CALL HCHAR(TARN+1,9,107)
910 NEXT TARN
920 KOB=2
930 FOR PLACE=30 TO 120 STEP
 30
940 KOB=KOB+1
950 RANDOMIZE :: YPLACE=INT(
RND*100+36)
960 IF (YPLACE/8)=INT(YPLACE
/8)THEN 970 ELSE 950
970 !
980 IF (YPLACE/16)<>INT(YPLA
CE/16)THEN 990 ELSE 950
990 CALL SPRITE(#KOB,104,2,Y
PLACE+1,57)
1000 FOR T=1 TO 10 :: NEXT T
 :: CALL COINC(ALL,TR)
1010 IF TR=-1 THEN 1020 ELSE
 1030
1020 CALL DELSPRITE(#KOB)::
GOTO 950
1030 NEXT PLACE
1040 CALL COINC(ALL,TR):: IF
 TR=-1 THEN 920
1050 FOR PLATZ=3 TO 15 STEP
3
1060 IND=PLATZ/3
1070 RANDOMIZE :: YP(IND)=IN
T(RND*3)+PLATZ :: RANDOMIZE
:: XP(IND)=INT(RND*5+1)
1080 CALL HCHAR(YP(IND),XP(I
ND),97):: NEXT PLATZ
1090 DISPLAY AT(2,17)SIZE(1)
:"0"
1100 DISPLAY AT(18,16)SIZE(3
):"180"
1110 DISPLAY AT(09,1)SIZE(1)
:"9"
1120 DISPLAY AT(10,1)SIZE(1)
:"0"
1130 DISPLAY AT(09,28)SIZE(2
):"2"
```

LISTINGS

```
1140 DISPLAY AT(10,28)SIZE(2
):"7"
1150 DISPLAY AT(11,28)SIZE(2
):"0"
1160 Y=95 :: X=220 :: CALL S
PRITE(#1,136,16,Y,X)
1170 DISPLAY AT(21,1):"" ::
DISPLAY AT(22,1):""
1180 RETURN
1190 !
1200 DISPLAY AT(19,1)SIZE(6)
:"HOEHE:"
1210 DISPLAY AT(19,13):"KOMP
:"
1220 DISPLAY AT(19,23):"SP:"
1230 DISPLAY AT(20,1):"VVERT
:"
1240 DISPLAY AT(20,13):"MACH
:"
1250 DISPLAY AT(21,1):"NEIG:
"
1260 DISPLAY AT(21,13):"D1:"
1270 DISPLAY AT(21,21):"D2:"
1280 DISPLAY AT(22,1):"NBRE:
"
1290 DISPLAY AT(22,13):"SCHU
B:"
1300 DISPLAY AT(23,1):"FWRK:
"
1310 DISPLAY AT(23,13):"FPS:
"
1320 DISPLAY AT(24,1):"TANK:
"
1330 DISPLAY AT(24,13):"DUEP
PEL:"
1340 RETURN
1350 !
1360 DISPLAY AT(19,7)SIZE(5)
:HOEHE
1370 DISPLAY AT(19,18)SIZE(4
):KOMP
1380 DISPLAY AT(19,26):SP
1390 DISPLAY AT(20,7)SIZE(5)
:VVERT
1400 DISPLAY AT(20,18):MACH
1410 DISPLAY AT(21,7)SIZE(4)
:CHR$(45);CHR$(NEIG);CHR$(45
)
1420 DISPLAY AT(21,16)SIZE(2
):D1
1430 DISPLAY AT(21,24):D2
1440 DISPLAY AT(22,7)SIZE(3)
:NBRE$
1450 DISPLAY AT(22,19):SCHUB
1460 DISPLAY AT(23,7)SIZE(3)
:FWRK$
1470 DISPLAY AT(23,17)SIZE(3
):FPS$
1480 DISPLAY AT(24,6)SIZE(6)
:TANK
1490 DISPLAY AT(24,21):DUEPPEL
```

```
1500 DISPLAY AT(1,1):""
1510 RETURN
1520 !
1530 CALL JOYST(1,SEIQU,VV)
1540 VV=-VV
1550 GOTO 1680
1560 CALL KEY(0,K,S)
1570 IF S=0 THEN 1650
1580 IF K=110 THEN 1980
1590 IF K<58 THEN 1600 ELSE
1610
1600 IF K>47 THEN 2210
1610 IF K=102 THEN 2310
1620 IF K=100 THEN 2430
1630 IF K=115 THEN 2490
1640 IF K=107 THEN 2360
1650 IF TANK<1 THEN SCHUB=0
1660 GOTO 2070
1670 RETURN
1680 !
1690 IF NBRE$="ON" THEN W=70
000 ELSE W=100000
1700 TANK=TANK-SCHUB/W
1710 ZW=121-NEIG
1720 IF HOEHE<=0 THEN 1730 E
LSE 1740
1730 IF MACH=0 THEN 1860 ELS
E 1820
1740 NEIG=NEIG-SEIQU/4
1750 IF NEIG>132 THEN NEIG=1
09
1760 IF NEIG<109 THEN NEIG=1
32
1770 KOM=KOM+15*COS(ZW/3.82-
PI/2)
1780 IF KOM>=360 THEN KOM=KO
M-360
1790 IF KOM<=0 THEN KOM=KOM+
360
1800 KOMP=KOM
1810 GOTO 1860
1820 KA=KA+(SEIQU*-2)
1830 IF LANDUNG THEN KA=KOMP
1840 IF KA>=360 THEN KA=KA-3
60
1850 IF KA<0 THEN KOMP=KA+36
0 ELSE KOMP=KA
1860 IF LANDUNG THEN 3050
1870 IF (VV=0)AND(HOEHE=0)TH
EN 1940
1880 IF MACH<MACHMIN THEN 18
90 ELSE 1930
1890 DISPLAY AT(1,1)BEEP:"..
..GESCHWINDIGKEIT !!"
1900 IF HOEHE>0 THEN 1910 EL
SE 1560
1910 VVERT=-5/MACH
1920 GOTO 1950
1930 HOEWI=SIN(ZW/3.82-PI/2)
:: VV=VV*HOEWI
1940 VVERT=MACH*VV/2+VVERT+H
OEWI-1
1950 HOEHE=HOEHE+2*VVERT
1960 EPOT=VVERT/30
1970 GOTO 1560
1980 !
1990 IF TANK>1 THEN 2010
2000 SCHUB=0 :: GOTO 2070
2010 IF NBRE$="OFF" THEN 202
0 ELSE 2050
2020 NBRE$="ON"
2030 SCHUB=SCHUB+(D1+D2)*278
2040 GOTO 2070
2050 NBRE$="OFF"
2060 SCHUB=SCHUB-(D1+D2)*278
2070 IF HOEHE>7000 THEN 2080
 ELSE 2100
2080 RANDOMIZE :: D1=INT(RND
*5)+1 :: RANDOMIZE :: D2=INT
(RND*5)+1
2090 SCHUB=(D1+D2)*500
2100 VZIEL=SCHUB/7000
2110 !
2120 IF HOEHE<=0 THEN VZIEL=
VZIEL/4
2130 IF VZIEL-EPOT=MACH THEN
 1670 ELSE 2140
2140 IF MACH>MACHMAX THEN 21
50 ELSE 2160
2150 VZIEL=MACHMAX :: DISPLA
Y AT(1,1)BEEP:"BELASTUNG WIR
D ZU GROSS !" :: REISS=REISS
+1
2160 IF VZIEL-EPOT>MACH THEN
 2170 ELSE 2190
2170 MACH=MACH+0.05
2180 GOTO 1670
2190 MACH=MACH-0.05
2200 GOTO 1670
2210 !
2220 IF TANK>1 THEN 2240
2230 D1=0 :: D2=0 :: SCHUB=0
 :: GOTO 2070
2240 IF HOEHE>7000 THEN RETU
RN
2250 PRESS=K-48
2260 SCHUB=PRESS*1000
2270 D1=PRESS
2280 D2=PRESS
2290 IF NBRE$="ON" THEN SCHU
B=SCHUB+(D1+D2)*278
2300 GOTO 2110
2310 !
2320 IF FWRK$="IN" THEN 2330
 ELSE 2340
2330 FWRK$="OUT" :: GOTO 167
0
2340 IF FWRK$<>"WEG" THEN FW
RK$="IN"
2350 GOTO 1670
2360 !
2370 IF FPS$="IN" THEN 2380
```

```
ELSE 2400
2380 FPS$="OUT" :: MACHMIN=0
.3 :: MACHMAX=0.8
2390 GOTO 1650
2400 IF FPS$<>"WEG" THEN 241
0 ELSE 1650
2410 FPS$="IN" :: MACHMIN=0.
6 :: MACHMAX=2.5
2420 GOTO 1650
2430 !
2440 IF DUEPPEL<1 THEN RETUR
N
2450 DUEPPEL=DUEPPEL-1
2460 IF ABS(Y1-YR)<12 THEN C
ALL DELSPRITE(#2)ELSE RETURN
2470 BG=0
2480 RETURN
2490 !
2500 IF (HOEHE=0)OR(SP=0)THE
N 1650
2510 SP=SP-1
2520 IF MACH>0.6 THEN 2530 E
LSE 2540
2530 DISPLAY AT(1,1)BEEP:"V
ZU GROSS : VERLUST NR.";SP+1
 :: GOTO 1650
2540 YVGL=INT((Y1+8)/8):: XV
GL=INT((X1+8)/8)
2550 FOR IND=1 TO 5
2560 IF SPRUNG(IND)=1 THEN 2
590
2570 IF (ABS(XVGL-XP(IND))<2
)AND(ABS(YVGL-YP(IND))<2)THE
N 2580 ELSE 2590
2580 SPRUNG(IND)=1 :: ORDEN=
ORDEN+1 :: CALL SOUND(100,30
0,5):: CALL SOUND(100,500,1)
:: CALL SOUND(100,400,8):: G
OTO 1650
2590 NEXT IND
2600 GOTO 1650
2610 GOTO 1670
2620 !
2630 YMOVE=-COS(KOMP/57.3)*M
ACH/2
2640 XMOVE=-SIN(KOMP/57.3)*M
ACH/2
2650 Y=Y+YMOVE :: X=X+XMOVE
2660 Y1=INT(Y):: X1=INT(X)
2670 IF FEHLER THEN RETURN
2680 CALL LOCATE(#1,Y1,X1)
2690 RETURN
2700 !
2710 IF (HOEHE>500)AND(X1>83
)AND(X1<155)THEN 2740
2720 CALL COINC(ALL,TR):: IF
 TR THEN 3180
2730 GOTO 2750
2740 RADAR=1
2750 IF VVERT THEN FLUG=1
2760 IF FLUG=0 THEN 2770 ELS
E 2790
2770 IF FWRK$="IN" THEN 3310
2780 IF (Y1>96)OR(Y1<80)OR(X
1<219)OR(X1>221)THEN 3340
2790 IF HOEHE<=0 THEN 2800 E
LSE 2820
2800 IF FLUG THEN 2810 ELSE
2830
2810 IF VVERT>-4 THEN 3020 E
LSE 3180
2820 IF FWRK$="WEG" THEN 285
0
2830 IF (FWRK$="OUT")AND(MAC
H>0.9)THEN 2840 ELSE 2850
2840 FWRK$="WEG" :: DISPLAY
AT(1,1)BEEP:"FAHRWERK ABGERI
SSEN"
2850 IF (Y<0)OR(X>256)OR(X<0
)OR(Y>142)THEN FEHLER=1 ELSE
 FEHLER=0
2860 IF FPS$="WEG" THEN 2890
2870 IF REISS>10 THEN 2880 E
LSE 2890
2880 MACHMIN=0.8 :: MACHMAX=
2.5 :: FPS$="WEG" :: DISPLAY
 AT(1,1)BEEP:"FLAPS ABGERISS
EN"
2890 IF (X1>180)AND(X1<200)T
HEN 3400 ELSE CALL DELSPRITE
(#2)
2900 RETURN
2910 VVERT=0 :: HOEHE=0
2920 CALL LOESCHEN
2930 DISPLAY AT(10,1):"ANERK
ENNUNG VERDIENT IHR....FLUG;
 ABER SIE HABEN IHRE...AUFGA
BE NICHT ERFUELLT."
2940 CALL GOON
2950 CALL LOESCHEN
2960 DISPLAY AT(7,1):"SIE HA
BEN DIE MISSION VOLL- STAEND
IG GEMEISTERT. IHRE...FLUGLE
ISTUNGSBEWERTUNG IST:";BEWER
TUNG;" PUNKTE"
2970 IF RADAR THEN 2980 ELSE
 3000
2980 DISPLAY AT(12,1):"DA SI
E UEBER DER HOEHE 200..GEFLO
GEN SIND, WEISS DER GE-GNER
UEBER DIE MISSION BE-..SCHEI
D."
2990 DISPLAY AT(16,1):"DESWE
GEN ERHALTEN SIE KEINENORDEN
!" :: GOTO 3010
3000 DISPLAY AT(16,1):"SIE E
RHALTEN EINEN ORDEN!"
3010 CALL GOON
3020 !
3030 IF LANDUNG THEN 3130
3040 IF MACH<0.5 THEN 3050 E
LSE 3280
```

```
3050 IF (Y1<96)AND(Y1>80)AND
(X1>216)AND(X1<224)THEN 3080
3060 IF (Y1<48)AND(Y1>40)AND
(X1>216)AND(X1<232)THEN 3080
3070 IF (Y1>112)AND(Y1<120)A
ND(X1>224)AND(X1<240)THEN 30
80 ELSE 3220
3080 IF LANDUNG THEN 1560
3090 IF FWRK$="OUT" THEN 310
0 ELSE 3250
3100 IF NEIG<>109 THEN 3370
3110 LANDUNG=1 :: HOEHE=0 ::
 VVERT=0 :: EPOT=0
3120 RETURN
3130 IF MACH>0 THEN RETURN
3140 DISPLAY AT(20,18):"0"
3150 CALL LOESCHEN
3160 BEWERTUNG=INT(ORDEN*(TA
NK*4+1))
3170 IF BEWERTUNG=0 THEN 292
0 ELSE 2950
3180 !
3190 CALL LOESCHEN
3200 DISPLAY AT(10,1):"VOR V
OLLENDUNG DER MISSION..SIND
SIE ABGESTUERZT! HOEHE, VVER
T UND HINDERNISSE SIND VON G
ROESSTER WICHTIGKEIT"
3210 CALL GOON
3220 CALL LOESCHEN
3230 DISPLAY AT(10,1):"SIE H
ABEN ES ZWAR GESCHAFFT,ZU ST
ARTEN, ABER SIE HABEN..DEN F
LUGPLATZ NICHT RICHTIG ANGEF
LOGEN UND GETROFFEN..."
3240 CALL GOON
3250 CALL LOESCHEN
3260 DISPLAY AT(10,1):"TROTZ
 IHRES SCHOENEN-FLUGES SOLLT
EN SIE NICHT DIE NERVENVERLI
EREN, UND VERGESSEN, AN DAS
FAHRWERK ZU DENKEN."
3270 CALL GOON
3280 CALL LOESCHEN
3290 DISPLAY AT(10,1):"DAS W
AR EIN ABSTURZ! ACHTEN SIE B
EI LANDUNGEN AUF IHRE..MACHZ
AHL, UND BEI FLUEGEN...AUF D
IE HOEHE."
3300 CALL GOON
3310 CALL LOESCHEN
3320 DISPLAY AT(10,1):"DAS F
AHRWERK WIRD VOR DEM...START
 NICHT EINGEFAHREN!"
3330 CALL GOON
3340 CALL LOESCHEN
3350 DISPLAY AT(10,1):"SIE S
OLLTEN VERSUCHEN, DIE..START
PISTE WAEHREND DES.....START
ENS NICHT ZU VERLASSEN!"
3360 CALL GOON
```

LISTINGS

```
3370 CALL LOESCHEN
3380 DISPLAY AT(10,1):"SIE D
UERFEN BEI DER LANDUNG NICHT
 SCHIEF DRINHAENGEN. ..DIE M
ISSION IST GESCHEITERT!"
3390 CALL GOON
3400 !
3410 IF BG<>1 THEN 3490
3420 IF (YR<10)OR(YR>140)OR(
XR<170)OR(XR>205)THEN 3430 E
LSE 3440
3430 BG=0 :: CALL DELSPRITE(
#22):: RETURN
3440 IF Y1>YR THEN YR=YR+2 E
LSE YR=YR-2
3450 IF X1>XR THEN XR=XR+2 E
LSE XR=XR-2
3460 CALL LOCATE(#2,YR,XR)
3470 CALL COINC(ALL,TR):: IF
 TR THEN 3180
3480 RETURN
3490 BG=1
3500 XR=190 :: YR=ABS(Y1-40)
+1
3510 CALL SPRITE(#2,92,2,YR,
XR)
3520 RETURN
3530 SUB LOESCHEN
3540 CALL DELSPRITE(ALL)
3550 CALL HCHAR(1,1,32,576)
3560 CALL CHARSET
3570 SUBEND
3580 SUB GOON
3590 DISPLAY AT(1,1):"DRUECK
EN SIE DEN FIRE-BUTTON"
3600 CALL KEY(1,K,S)
3610 IF (S=0)OR(K<>18)THEN 3
600
3620 CALL CLEAR
3630 DISPLAY AT(10,1):"CHECK
 YOUR SYSTEMS, WE' RE..GOING
 TO TAKE OFF AT ONCE !"
3640 RUN 120
3650 SUBEND
3660 SUB START
3670 CALL CLEAR
3680 DISPLAY AT(10,1)BEEP:"W
ICHTIG:ALPHA LOCK ENTRASTEN"
3690 PRINT "BEWEGEN SIE IHRE
N JOYSTICK..NACH OBEN!"
3700 CALL JOYST(1,Y,X)
3710 IF X=4 THEN 3720 ELSE 3
700
3720 CALL CLEAR
3730 SUBEND
3740 SUB EINLEITUNG
3750 DISPLAY AT(9,1)ERASE AL
L:".....FLUGSIMULATOR 1.7...
.............BY.............
....KARSTEN KNOBEL,NEUSTADT/
W "
```

```
3760 DISPLAY AT(15,1):".....
.......ODER.................
.DIE FUENF SPIONE......AUS D
EM LAND DER NEUGIERIGEN"
3770 PRINT "COPYRIGHTS BY S.
O."
3780 PRINT "DRUECKEN SIE EIN
E TASTE !"
3790 CALL CHAR(96,"000000000
1030700",97,"00000000E0FCFF1
8",98,"0000000000000C000",99,
"0000000000000003")
3800 CALL CHAR(100,"00008081
9FBFBFFF",101,"28488405FFFFF
FFF",102,"00589808FFFFFFFF",
103,"070F1FFFFFFFFFFF")
3810 CALL CHAR(104,"BFBE9C83
80030703",105,"FF3F00FF8080C
080",106,"FFFF3EF000000000",
107,"00183E7C7C200000")
3820 BG=95
3830 FOR ZEILE=2 TO 4
3840 FOR SPALTE=14 TO 17
3850 BG=BG+1 :: CALL HCHAR(Z
EILE,SPALTE,BG):: IF BG=106
THEN BG=31
3860 NEXT SPALTE
3870 NEXT ZEILE
3880 CALL KEY(0,K,S):: IF S<
>0 THEN 3950
3890 FOR S=1 TO 5
3900 FOR T=1 TO 14 :: CALL K
EY(0,K,VV):: IF VV<>0 THEN 3
950
3910 NEXT T :: CALL SOUND(-2
000,-4,0)
3920 CALL MAGNIFY(1):: CALL
SPRITE(#S,107,16,25,110,1,14
)
3930 NEXT S
3940 GOTO 3880
3950 CALL DELSPRITE(ALL)
3960 PRINT "FLUGBLATT (WICHT
IGE DATEN) ": : :
3970 PRINT "FLAPS=OUT -MACHM
IN=0.3 ..............-MACHM
AX=0.8 .....FLAPS=IN..-MACHM
IN=0.6 ..............-MACHM
AX=2.5"
3980 PRINT "FLAPS=WEG -MACHM
IN=0.8 ..............-MACHM
AX=2.5": :
3990 PRINT "DER ABWURF EINES
 SPIONES ER-FOLGT BEI MACH<=
0.55 UND IM-MER AUF EIN ANDE
RES ZIEL": :
4000 PRINT "DIE LANDUNG ERFO
LGT BEI.....MACH<0.5 UND VVE
RT > -4"
4010 PRINT "DER REST ERGIBT
SICH VON....SELBST !": : : :
4020 PRINT "..--HOLM UND RIP
PENBRUCH--.....DRUECKEN SIE
<ENTER> !"
4030 CALL KEY(0,K,S):: IF S=
0 THEN 4030
4040 IF K=13 THEN 4050 ELSE
4030
4050 SUBEND
```

LISTINGS

Fortsetzung von Seite 58

FLUGSIMULATOR

chen Höhen entdeckt wird (beim Entdecken durch das Radar bekommt man keine Punktbewertung).
In 2810 kann der Wert der Sinkgeschwindigkeit dem Können des Piloten angeglichen werden.
In 2870 kann die Vorwarnzeit für das Abreißen der Flaps eingestellt werden.
In 3040 läßt sich die maximale Machzahl bei den Landungen verändern.
Das gleiche läßt sich in 2520 bei der Abwurfgeschwindigkeit der Spione auf die Ziele einstellen.
In 3500 kann man den Startpunkt der Raketen variieren, wobei y1/x1 den Standpunkt des Flugzeugs beschreibt.
Die jetzt eingestellten Werte sind jedoch sinnvoll bzw. realistisch...
"Höhe" zeigt die Höhe über NN an, nicht über Grund (Berge!).
"vvert" (v vertikal) zeigt Steigen und Sinken an.
"Mach" bezieht sich nur auf die horizontale Komponente der Geschwindigkeit.
"D1" und "D2" regeln primär den Schub, der Nachbrenner gibt darüberhinaus einen prozentualen Anteil dazu. Mit dem Nachbrenner fliegt man aber deutlich unwirtschaftlicher, was sich vorzeitig in einem leeren Tank äußern kann.
"Fwrk" zeigt den Lagezustand des Fahrwerks an. Bei nicht eingefahrenem Zustand reißt es bei Mach 0.9 ab.
"Sp" zeigt die Anzahl der verbleibenden Spione an.
"Fps" zeigt an, ob die Klappen ein- oder ausgefahren sind und regelt minimale bzw. maximale Fluggeschwindigkeit.
"Dueppel" zeigt an, wie oft man noch düppeln kann.
Der "Tank" informiert über die restliche Spritmenge.
Der "Kompaß" (komp) legt den Kurs fest. Das geschieht so: Man stellt sich zunächst einen Kreis um das Flugzeug vor. Das Flugzeug bildet den Anfangspunkt eines Pfeils und das Pfeilende zeigt auf die gedachte Gradmarkierung am Kreis (vgl. mit Gradzahl am Bildschirmrand). Das Flugzeug wird dem Pfeil nachfliegen.
Beispiel: Zeigt der Kompaß die Zahl 90 an, so wird das Flugzeug von seiner derzeitigen Position aus waagerecht nach links fliegen.
Es handelt sich also eigentlich um eine ganz normale Kompaßnavigation.
Anmerkung: Je größer Mach, desto schneller bewegt sich natürlich das Flugzeug auf seinem Kurs.
Der Neigungswinkel dient zum Erkennen, wann man gegenlenken muß, um die angestrebte Kompaßeinstellung zu erreichen.
Er stellt keinen künstlichen Horizont dar!
"Neig,Komp,vvert und Höhe" werden mit dem Joystick angesteuert. Die übrigen Funktionen werden mit der Tastatur angesteuert. Drückt man auf der Tastatur den Anfangs-

buchstaben der ausgewählten Funktion, so hat das eine Änderung des derzeitigen Zustands zur Folge, wobei man beachten muß, daß die Flaps über "k" wie Klappen angelenkt werden, weil das "f" bereits für das Fahrwerk reserviert ist. Spione und Düppel werden bei jedem Aufruf immer um eins vermindert.
Tip: Solange auf dem Joystick oder der Taste bleiben, bis sich der Zustand ändert.
So sieht ein Bilderbuchflug aus:
Um das Flugzeug zu starten, wird zuerst der Maximalschub eingestellt (Nachbrenner nicht vergessen!). Nun werden die Flaps ausgefahren, um die minimale Fluggeschwindigkeit zu ermöglichen. Beim Erreichen dieser Geschwindigkeit bewirkt ein Ziehen am Joystick das Abheben des Flugzeugs. Das Fahrwerk wird sinnvollerweise jetzt eingefahren. Wenn Mach >0.6 ist, werden die Klappen eingefahren. Jetzt wird – unter Berücksichtigung des Neigungswinkels – eine Kurve eingeleitet, wobei bereits auf eine Lücke in den Bergen geachtet wird.
Anmerkung: Rollen und Rückenflug sind möglich, aber keine Loopings. Beim Rückenflug dreht sich die Joystickfunktion natürlich um, und in Kurven wird der Wirkungsgrad des Höhen/Tiefenruders schlechter. Wenn man die Grenze überflogen hat, wird man von Raketen attackiert, vor denen man sich nur dürch Düppeln schützen kann. Man wartet solange, bis die Raketen einen fast berühren, dann düppelt man (Taste "d" drücken!). Danach muß man entweder die Bergtäler – die Zwischenräume – durchfliegen, oder man fliegt in einer größeren Höhe als 500 über diese hinweg, wobei man aber keine Chance mehr hat, eine positive Flugbewertung zu erhalten.
Jetzt muß man durch die Flak fliegen. Sie ist zwar nicht überall besetzt, aber man sollte doch besser die Lücken treffen. Deshalb – zur besseren Orientierung – sind auch die ganzen Punkte auf dem Bildschirm. Außerdem kann die Flak nicht überflogen werden.
Nun müssen die Ziele mit Mach <=0.6 angeflogen werden. Genau im Zentrum läßt man einen Spion abspringen. Dann werden die anderen Spione auf die anderen Ziele abgeworfen, und man fliegt zurück.
Die Landung ist recht schwierig und sollte deshalb recht großräumig auf einer der drei Landebahnen angesetzt werden. Die Landung erfolgt mit Mach <=0.5, und die Sinkgeschwindigkeit muß kleiner als 4 sein (vvert >–4). Wenn Sie gelandet sind, müssen die Düsen noch ausgeschaltet werden. Erst nachdem die Rollgeschwindigkeit 0 ist, erscheint die Flugbewertung. Die Feinheiten des Programms werden Sie selbst herausfinden, nicht zuletzt, weil es in der Übungsphase zu diversen kunstgerechten Abstürzen kommen wird.
Zu erwähnen bleibt noch, daß die Geschwindigkeit des Programms natürlich nicht mit Maschinensprache zu vergleichen, aber daß sie trotz der vielen Berechnungen und Vergleiche noch vertretbar ist. Das ist möglicherweise sogar ein gewisser Vorteil für den Uneingeweihten. *Karsten Knobel*

TI REVUE:
Die Nummer 1 in Europa!

ZAUBER-SCHLOSS

Ziel dieses Grafik-Adventure-Spiels ist es, in dem Schloß einen Schatz zu finden und anschließend wohlbehalten nach draußen zu entkommen.
Das Programm besteht aus insgesamt vier Teilen (LOAD, ANFANG, START, PORTAL), die unter den im Kopf des jeweiligen Listings angegebenen Namen abgespeichert werden sollten, da sie sich untereinander automatisch starten.
Beim Eintippen des Programms sind besonders die Zeilen 210 – 240 des zweiten Teils und die letzten neun Zeilen des vierten Teils zu beachten: Hier werden Characternummern über 127 angesprochen. CTRL bedeutet jeweils keine einzugebende Buchstabenfolge, sondern die CONTROL-Taste.
Alle Buchstaben in geschweiften Klammern sind zusammen mit der Control-Taste einzugeben, die Klammern selbst sind nicht einzugeben! Allgemein: Groß- und Kleinbuchstaben beachten! Der Sprachsynthesizer ist nicht unbedingt notwendig, aber dann sollten alle Sprachausgabebefehle mit DISPLAY AT unten im Bildschirm angezeigt werden, gefolgt von einer kurzen Zeitschleife. Dies ist jedoch nur wenig Arbeit.
Einige Hinweise auf die Lösung des Adventure werden Sie sicher schon beim Eintippen entdecken, da die Befehlseingabeverarbeitung nicht verschlüsselt ist. Dennoch ist die Lösung nicht einfach.
Einige Hinweise sollen den Einstieg ins Spiel etwas erleichtern: Im demjenigen Stockwerk des Schlosses, in dem sich der Spieler zu Beginn des Spieles befindet, kann mit der Eingabe 'HILFE' eine Liste von Verben, die als Eingabe akzeptiert werden, eingesehen werden (kein Anspruch auf Vollständigkeit!). Ebenso werden alle Abkürzungen von Wörtern angezeigt, die so und nicht anders eingegeben werden müssen: Wortteile zwischen Bindestrichen müssen bei der Befehlseingabe weggelassen werden, z.B. VERL-IERE bedeutet, daß der Befehle 'verliere' als 'VERL' eingegeben werden muß. Der Parser (Befehlseingabeverarbeitungs-Programmteil) ist nicht flexibel. Es sind nur Zweiwortbefehle zugelassen, außer bei Richtungsangaben. Dabei sind nur die Abkürzungen N, S, O, W sowie HOCH und RUNTER einzugeben.
Führen Sie gut Buch über die Gegenstände, die Sie mit sich führen, denn in jedem Stockwerk gibt es nur einen Raum, in dem man mit dem Befehl 'WARTE' die mitgeführten Sachen auflisten kann. Man kann immer nur zwei Gegenstände mitführen. Überflüssige oder teilverbrauchte Dinge können nur in dem Raum wieder abgelegt werden, in dem sie aufgenommen wurden! Daher: Von Anfang an genaue Karte anlegen.
Leider bietet das Programm keine Möglichkeit, das Spiel zwischendurch zu speichern, so daß nach jedem Abbruch ganz von vorn begonnen werden muß. Das kostet manchmal ein paar Minuten, wird aber den echten Adventurefreak sicher nicht von diesem Spiel abhalten.

```
10 ! ************************
11 ! *   ZAUBERSCHLOSS      *
12 ! *  (Teil 1: "LOAD")    *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Thorsten Ziehm     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *       32K-Erw.       *
23 ! *    Speech-Synth.     *
24 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *      375 Bytes       *
28 ! *                      *
29 ! ************************
100 FOR A=0 TO 8 :: CALL COL
OR(A,9,1):: NEXT A
110 CALL CLEAR
120 CALL SCREEN(16)
130 DISPLAY AT(1,1):"....TEX
AS..INSTRUMENTS "
140 CALL SAY("#TEXAS INSTRUM
ENTS")
150 DISPLAY AT(10,1):"......
..T. I. S. S. ..............
.............................
...presents:"
160 DISPLAY AT(16,1):"......
ZAUBERSCHLOSS"
170 FOR A=1 TO 999 :: NEXT A
180 CALL CLEAR :: DISPLAY AT
(12,1):".........LOADING..."
190 RUN "DSK1.ANFANG"
```

LISTINGS

```
10 ! ************************
11 ! *   ZAUBERSCHLOSS      *
12 ! * (Teil 2: "ANFANG")   *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Thorsten Ziehm     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *       32K-Erw.       *
23 ! *    Speech-Synth.     *
24 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     10066 Bytes      *
28 ! *                      *
29 ! ************************
100 !CALL INIT :: CALL LOAD(
-31806,16):: CALL LOAD(-3193
1,66)
110 CALL SCREEN(2)
120 CALL CHAR(140,"444444444
4444444",129,"444444FFFF4444
44")
130 CALL CHAR(130,"4444EEEE4
4440000",131,"C44464544C4446
45",132,"C44464FFFF444645",1
33,"C444EEFE4C440201",134,"4
546EEEE54644080")
140 CALL CHAR(135,"4546444C5
46444C4",136,"454644FFFF6444
C4",137,"FFC5C5C5C5FFC5C5",1
38,"C5C5C5C5C5CDC5C5",139,"C
5C5C5FFFFC5C5FF")
150 !
160 CALL COLOR(8,11,1,13,11,
1)
170 CALL COLOR(14,11,1,2,16,
1,3,16,1,4,16,1,5,3,1,6,3,1,
7,3,1,8,3,1,9,3,1,10,3,1,11,
3,1)
180 A$(1)="a a a a.........
...a a a a aaaaaaa.........
...aaaaaaa aaaaaaa.........
...aaaaaaa aaaaaaa.........
...aaaaaaa "
190 A$(2)="aaaaaaa.........
...aaaaaaa aaaaaaa.........
...aaaaaaa aaaaaaa a a a a a
 a aaaaaaa aaaaaaaaaaaaaaaaa
aaaaaaaaaa "
200 A$(3)="aaaaaaaaaaaaaaaaa
aaaaaaaaaa aaa^_aaaaaaaaaaaa
aaaaa^_aaa aaagfaaaaaaaaaaaa
aaaaagfaaa aaadeaaaaaaaaaaaa
aaaaadeaaa "
210 A$(4)="aaaaaaaaaaaaaaaaa
aaaaaaaaaa aaaaaaaaaaaaaaaaa
aaaaaaaaaa aaaaaaaaaccccccc
```

```
caaaaaaaaa aaaaaaaaabCTRL(CL
LLLLG)baaaaaaaaa "
220 A$(5)="aaaaaaaaabCTRL(LC
LLLGL)baaaaaaaaa aaaaaaaaabC
TRL(AADAHAA)baaaaaaaaa aaaaa
aaaabCTRL(LLLILLL)baaaaaaaaa
 aaaaaaaaabCTRL(LLLJLLL)baaa
aaaaaa "
230 A$(6)="aaaaaaaaabCTRL(AA
AKAAA)baaaaaaaaa aaaaaaaaabC
TRL(LLGLCLL)baaaaaaaaa aaaaa
aaaabCTRL(LGLLLCL)baaaaaaaaa
 aaaaaaaaabCTRL(FBBBBBE)baaa
aaaaaa "
240 CALL CHAR(42,"000000000000
000000",43,"FFFFFFFFFFFFFFFF
",47,"FF818181818181FF",97,"
FDFDFD00DFDFDF00",98,"5A5A5A
5A5A5A5A5A")
250 CALL CHAR(99,"FF00DD5577
00FFFF",100,"FFC1C1C1C1C1FFF
F",101,"FF8383838383FFFF",10
2,"FF838383838383FF",103,"FF
C1C1C1C1C1C1FF")
260 CALL CHAR(94,"FFF9F1E1C1
C1C1FF",95,"FF9F8F87838383FF
",106,"8040201008040201",107
,"0102040810204080",108,"FFC
0A09088848281")
270 CALL CHAR(109,"FF0305091
1214181",110,"8182848890A0C0
FF",111,"8141211109050 3FF",1
12,"FF0000000000000000",113,"0
000000000000000FF")
280 CALL CHAR(114,"8080808080
8080808",115,"0101010101010101
",117,"FF818181FF818181",1
18,"818181818D818581",119,"8
1818181818181FF")
290 CALL CHAR(120,"818181818
18181FF")
300 DISPLAY AT(1,1):A$(1);A$
(2);A$(3);A$(4);A$(5);A$(6)
310 !
320 !
330 RANDOMIZE
340 !---TITELBILD---
350 !
360 CALL HCHAR(2,10,32,13)::
 CALL HCHAR(4,12,32,10)
370 CALL CHAR(62,"000000000000
000000"):: CALL COLOR(10,11,
1,11,11,1,12,11,1)
380 CALL CHAR(88,"0202020202
020202"):: CALL VCHAR(3,17,8
8,4)
390 CALL SAY("IF+YOU+WANT+TO
+PLAY,YOU+MUST+PRESS J, IF+Y
OU+DO+NOT+WANT,PRESS N")
400 CALL KEY(0,K,S):: IF S=0
 THEN 400 :: IF K=78 OR K=11
0 THEN 1200
410 FOR TY=24 TO 16 STEP -1
:: CALL SOUND(-100,-4,0):: C
ALL HCHAR(TY,13,32,7):: ON T
Y-15 GOSUB 1110,1120,1130,11
40,1150,1160,1170,1180,1190
:: NEXT TY
420 CALL COLOR(13,10,1)
430 CALL CHAR(120,"00F0FFFF0
F0000F0FFFF0F0000000000023EF
EFEC2020 23EFEFEC2020202020 2"
)
440 CALL MAGNIFY(3)
450 !
460 CALL SPRITE(#1,120,10,40
,121,-2,0):: FOR SD=1 TO 600
 :: NEXT SD
470 CALL MOTION(#1,0,0)
480 CALL SAY("#READY TO STAR
T"):: GOTO 1220
490 FOR SD=1 TO 500 :: NEXT
SD :: CALL CLEAR :: CALL DEL
SPRITE(ALL)
500 !
510 !---CHARS UND EINFAERBUN
GEN---
520 !
530 !
540 !
550 CALL SCREEN(2):: FOR A=1
 TO 2 :: CALL COLOR(A,9,1)::
 NEXT A :: CALL COLOR(3,15,1
)
560 FOR A=4 TO 13 :: CALL CO
LOR(A,4,1):: NEXT A
570 CALL CHAR(93,"000000000000
003030")!** PUNKT (.=])
580 CALL CHAR(94,"000000000003
0102 00")!** KOMMA (,=^)
590 CALL CHAR(91,"0000303000
303000")!** DOPPELPUNKT(:=[)
600 CALL CHAR(92,"0038440408
100010")!** FRAGEZ.(?=\)
610 !---WACHE,GIFT,usw.---
620 CALL CHAR(60,"0000000304
0808091010090601000000E3141
C122A141222C109020C0000000")
!** STEIN
630 CALL CHAR(56,"8080707878
3C0E0703000000000000000000000
000000000E0F03818181C0E0601")
!** SCHLANGE
640 CALL CHAR(52,"08401F3F68
EEB3B8BFDFF98FE97B3B9F800490
E0A0B195F8FC94B5F4FCF2E084")
!** WOLKE
650 CALL CHAR(140,"0B0607030
103071F3F2F2F7753010107E8B0F
060C0E0F0FCFEFAFAF7E5C04070"
)!** KOBOLD
```

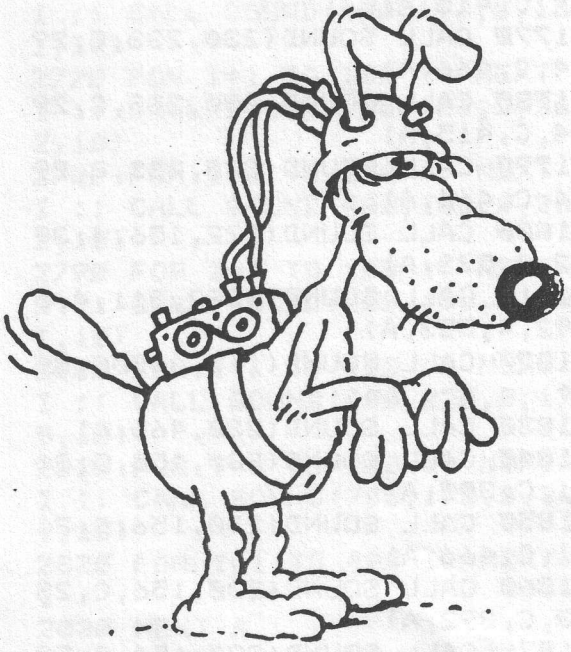
```
660 CALL CHAR(100,"000003010
1030604040404040404070000E
0C0C0603010101010101010F0"
)!** FLASCHE (GIFT)
670 CALL CHAR(104,"000000000
0000000003020302010203000000
0000000000000C0E0A0C0E0A000"
)!** INHALT (GIFT)
680 CALL CHAR(108,"00FC84B4B
784FC0000000000000000040300
8304080000000000000000000000"
)!** SCHALTER (TUER)
690 CALL CHAR(112,"000000000
0000000000FF0000000000000000
0000000000804FE040800000000"
)!** SPEER
700 CALL CHAR(116,"000000000
0000000FFFAB020000000000000
000001C3662C262361C00000000"
)!** SCHLUESSEL
710 CALL CHAR(120,"00081C1C2
A08191F1F0909090809090B60F0F
0F0F060F87CFEFEFAFAF298989C"
)!** WACHE
720 CALL CHAR(124,"000000000
0000000000000000000000000000
0204040E0E0E0E0E0E0E0E0E0E0"
)!** DYNAMIT
730 CALL CHAR(96,"000000000000
00000070A143F2828283F000000
0000000000FC2C54E4A4A8B0E0")
!** KISTE (SCHATZ)
740 CALL CHAR(136,"0000007F0
05755755557007F000000000000
0FF00754547457500FF00000000"
)!** HOCH
750 CALL CHAR(130,"7F007A4A7
A524B00007D11111111007FFF005
1595553D10000EF09CF0AE900FF"
)!** RUNTER
760 CALL CHAR(95,"000000F0F0
")!** BINDESTRICH(-=_)
770 CALL CHAR(33,"405C505C50
5C4054544854544048484848C840
5C48484848")!** EXIT (RAUM U
)
780 CALL CHAR(36,"0202020202
0203020280A8AAAAAFA0202020202
0202020202")!** WEST-TUER
790 CALL CHAR(39,"02FF000000
0000000000FF007C407C047C00FF00
0000000000")!** SUED-TUER
800 CALL CHAR(42,"000000000000
00FF004464544C4400FF00000000
000000FF40")!** NORD-TUER
810 CALL CHAR(45,"4040404040
4040404005F5151515F404040C040
4040404040")!** OST-TUER
820 CALL CHAR(134,"FFABFFD5F
FABFFD5")!** MAUER (RAUM U)
830 CALL CHAR(48,"E0E0E0E0E0
E0E0E0"):: CALL CHAR(49,"070
707070707070707"):: CALL CHAR(
50,"FFFFFF"):: CALL CHAR(51,
"000000000000FFFFFF")!** SEITE
NWAENDE
840 !
850 CHAR$="1253678465F34536"
860 CALL COLOR(5,16,1,6,16,1
,7,16,1,8,16,1,0,4,1,14,11,1
,13,8,1)
870 !
880 CALL MAGNIFY(3)
890 DISPLAY AT(1,1):"Z A U B
 E R S C H L O S S "
900 DISPLAY AT(9,1):"......C
OPYRIGHT..BY" :: DISPLAY AT(
12,1):".........TIIJSJSJ"
910 L$="TOP"
920 B$="START"
930 C$="ENDS"
940 D$="SAY"
950 E$="E"
960 F$="MIGHT"
970 CALL SPGET(L$,AA$)
980 AAA$=SEG$(AA$,1,35)
990 CALL SPGET(B$,BB$)
1000 BBB$=SEG$(BB$,1,20)
1010 CALL SPGET(C$,CC$)
1020 CCC$=SEG$(CC$,1,66)
1030 CALL SPGET(D$,DD$)
1040 DDD$=SEG$(DD$,1,22)
1050 CALL SPGET(F$,FF$)
1060 FFF$=SEG$(FF$,1,18)
1070 CALL SPGET(E$,EE$)
1080 EEE$=SEG$(EE$,1,40)
1090 CALL SAY(,AAA$,,BBB$,,C
CC$,,DDD$,,EEE$,,FFF$)
1100 FOR T=1 TO 200 :: NEXT
T :: RUN "DSK1.START"
1110 DISPLAY AT(16,11)SIZE(7
):"j.....k" :: RETURN
1120 DISPLAY AT(17,11)SIZE(7
):" j...k " :: RETURN
1130 DISPLAY AT(18,11)SIZE(7
):"..j k.." :: RETURN
1140 DISPLAY AT(19,11)SIZE(7
):"...u..." :: RETURN
1150 DISPLAY AT(20,11)SIZE(7
):"...v..." :: RETURN
1160 DISPLAY AT(21,11)SIZE(7
):"...w..." :: RETURN
1170 DISPLAY AT(22,11)SIZE(7
):"..k j.." :: RETURN
1180 DISPLAY AT(23,11)SIZE(7
):" k...j " :: RETURN
1190 DISPLAY AT(24,11)SIZE(7
):"k.....j" :: RETURN
1200 CALL INIT :: CALL LOAD(
-31806,0):: CALL LOAD(-31931
,0):: CALL LOAD(-31804,0,36)
```

```
1210 ! LIED
1220 ! RAINBOW*TI BASIC
1230 GOSUB 1330
1240 CALL SOUND(230,40000,30
)
1250 GOSUB 1330
1260 CALL SOUND(115,40000,30
)
1270 GOSUB 1630
1280 CALL SOUND(10,40000,30)
1290 GOSUB 1330
1300 CALL SOUND(1000,40000,3
0)
1310 !
1320 GOTO 490
1330 A=0 :: B=6 :: C=9
1340 CALL SOUND(922,196,B,23
3,B,311,A)
1350 CALL SOUND(922,392,B,46
6,B,622,A)
1360 CALL SOUND(461,294,B,46
6,B,587,A)
1370 CALL SOUND(230,294,B,39
2,B,466,A)
1380 CALL SOUND(230,294,B,44
0,B,523,A)
1390 CALL SOUND(461,294,B,46
6,B,587,A)
1400 CALL SOUND(461,277,B,52
3,B,622,A)
1410 CALL SOUND(922,208,B,26
2,B,311,A)
1420 CALL SOUND(461,156,B,39
2,B,523,A)
1430 CALL SOUND(461,156,B,37
0,B,523,A)
1440 CALL SOUND(461,196,B,34
9,B,466,A)
1450 CALL SOUND(461,196,B,31
1,B,466,A)
1460 CALL SOUND(461,175,B,29
4,B,466,A)
1470 CALL SOUND(461,165,B,27
7,B,466,A)
1480 CALL SOUND(922,156,B,20
8,B,262,A)
1490 CALL SOUND(922,175,B,31
1,B,415,A)
1500 CALL SOUND(461,233,B,31
1,B,392,A)
1510 CALL SOUND(230,233,B,26
2,B,311,A)
1520 CALL SOUND(230,233,B,29
4,B,349,A)
1530 CALL SOUND(461,233,B,33
0,B,392,A)
1540 CALL SOUND(461,233,B,27
7,B,415,A)
1550 CALL SOUND(461,220,B,29
4,B,349,A)
```

LISTINGS

```
1560 CALL SOUND(230,220,B,24
7,B,294,A)
1570 CALL SOUND(230,220,B,26
2,B,311,A)
1580 CALL SOUND(461,208,B,29
4,B,349,A)
1590 CALL SOUND(461,208,B,29
4,B,392,A)
1600 CALL SOUND(1383,196,B,2
33,B,311,A)
1610 CALL KEY(0,K,S):: IF S<
>0 THEN 490
1620 RETURN
1630 CALL SOUND(230,466,A)
1640 CALL SOUND(230,156,C,31
1,C,392,A)
1650 CALL SOUND(230,156,C,31
1,C,466,A)
1660 CALL SOUND(230,156,C,23
3,C,392,A)
1670 CALL SOUND(230,156,C,23
3,C,466,A)
1680 CALL SOUND(230,156,C,26
2,C,392,A)
1690 CALL SOUND(230,156,C,26
2,C,466,A)
1700 CALL SOUND(230,156,C,23
3,C,392,A)
1710 CALL SOUND(230,156,C,23
3,C,466,A)
1720 CALL SOUND(230,262,C,31
1,C,415,A)
1730 CALL SOUND(230,262,C,31
1,C,466,A)
1740 CALL SOUND(230,262,C,31
1,C,415,A)
1750 CALL SOUND(230,262,C,31
1,C,466,A)
1760 CALL SOUND(230,233,C,29
4,C,415,A)
1770 CALL SOUND(230,233,C,29
4,C,466,A)
1780 CALL SOUND(230,233,C,29
4,C,415,A)
1790 CALL SOUND(230,233,C,29
4,C,466,A)
1800 CALL SOUND(922,156,4,39
2,4,523,A)
1810 CALL SOUND(1382,311,4,3
92,4,523,A)
1820 CALL SOUND(115,40000,30
)
1830 CALL SOUND(230,466,A)
1840 CALL SOUND(230,156,C,31
1,C,392,A)
1850 CALL SOUND(230,156,C,31
1,C,466,A)
1860 CALL SOUND(230,156,C,23
3,C,392,A)
1870 CALL SOUND(230,156,C,23
3,C,466,A)
```

```
1880 CALL SOUND(230,156,C,26
2,C,392,A)
1890 CALL SOUND(230,156,C,26
2,C,466,A)
1900 CALL SOUND(230,156,C,23
3,C,392,A)
1910 CALL SOUND(230,156,C,23
3,C,466,A)
1920 CALL SOUND(230,262,C,31
1,C,440,A)
1930 CALL SOUND(230,262,C,31
1,C,523,A)
1940 CALL SOUND(230,262,C,31
1,C,440,A)
1950 CALL SOUND(230,262,C,31
1,C,523,A)
1960 CALL SOUND(230,185,C,31
1,C,440,A)
1970 CALL SOUND(230,185,C,31
1,C,523,A)
1980 CALL SOUND(230,185,C,31
1,C,440,A)
1990 CALL SOUND(230,185,C,31
1,C,523,A)
2000 CALL SOUND(922,349,4,46
6,4,587,A)
2010 CALL SOUND(922,311,4,37
0,4,587,A)
2020 CALL SOUND(922,349,4,52
3,4,698,A)
2030 CALL SOUND(922,294,4,37
0,4,523,A)
2040 CALL KEY(0,K,S):: IF S<
>0 THEN 4990
2050 RETURN
```



LISTINGS

```
10 ! ***********************
11 ! *   ZAUBERSCHLOSS     *
12 ! * (Teil 3: "START")   *
13 ! *                     *
14 ! *    Copyright by     *
15 ! *                     *
16 ! *   Thorsten Ziehm    *
17 ! *                     *
19 ! * Benoetigte Geraete  *
20 ! *  TI99/4A  Konsole   *
21 ! *     Ext. Basic      *
22 ! *       32K-Erw.      *
23 ! *    Speech-Synth.    *
24 ! *                     *
26 ! *  Speicherbelegung   *
27 ! *    15923 Bytes      *
28 ! *                     *
29 ! ***********************
100 ON BREAK NEXT
110 KOBOLD$="DA" :: RING$="D
A" :: DABEI=0 :: MAUER$="DA"
 :: F=1 :: I=1 :: GIFT$="DA"
 :: SCHLUESSEL$="DA" :: WACH
E$="DA" :: SPEER$="DA" :: DY
NAMIT$="DA"
120 KAMMER$="ZU" :: KISTE$="
DA" :: STEIN$="DA" :: SCHLAN
GE$="DA"
130 !
140 CALL MAGNIFY(3):: GOTO 1
6220
150 ! A
160 IF GIFT$="DABEI" THEN 18
0 ELSE TE$="GEGENSTAENDE[...
.....
.....GIFT"
170 CALL SPRITE(#1,100,2,60,
60,0,0,#2,104,2,60,60)
180 CALL ZIMMER(A$,1,1,0,1,T
E$,15,14)
190 TE$=""
200 BEF=BEF+1 :: CALL DELSPR
ITE(ALL):: IF A$="HILFE" THE
N CALL HILFE :: GOTO 160 ELS
E 210
210 IF A$="NIMM GIFT" AND DA
BEI<3 AND GIFT$="DA" THEN CA
LL SAY("O+K"):: AA$="GIFT" :
: DABEI=DABEI+1 :: GIFT$="DA
BEI" :: CALL DELSPRITE(ALL)
220 IF A$="VERL GIFT" AND GI
FT$="DABEI" THEN GIFT$="DA"
:: DABEI=DABEI-1 :: CALL SAY
("O+K"):: AA$=""
230 IF A$="TRINKE GIFT" THEN
 240 ELSE IF A$="O" THEN 250
 ELSE IF A$="N" OR A$="W" TH
EN 2420 ELSE 160
240 TEXT$(1)="SIE HABE^ DAS
GIFT GETRUNKEN.....
```

```
.............UND SIND DARAN
ØGESTORBEN." :: CALL ENDE(TE
XT$())

25Ø ! B
26Ø CALL ZIMMER(A$,Ø,Ø,1,1,"
",1,1)
27Ø BEF=BEF+1 :: IF A$="HILF
E" THEN CALL HILFE :: GOTO 2
6Ø ELSE 28Ø
28Ø IF A$="W" THEN 15Ø ELSE
IF A$="S" THEN 162Ø ELSE 26Ø
29Ø ! C
3ØØ IF STEIN$<>"DA" THEN 32Ø
31Ø TE$="GEGENSTAENDE[.....
.............................
.....STEIN" :: CALL SP
RITE(#1,6Ø,2,7Ø,5Ø)
32Ø CALL ZIMMER(A$,Ø,Ø,Ø,Ø,T
E$,15,Ø)::
33Ø TE$=""
34Ø BEF=BEF+1 :: CALL DELSPR
ITE(#1):: IF A$="NIMM STEIN"
 AND STEIN$="DA" AND DABEI<2
 THEN STEIN$="DABEI" :: CALL
 SAY("O+K"):: DABEI=DABEI+1
:: FF$="STEIN"
35Ø IF A$="VERL STEIN" AND S
TEIN$="DABEI" THEN CALL SAY(
"O+K"):: DABEI=DABEI-1 :: FF
$="" :: STEIN$="DA"
36Ø !
37Ø IF A$="HOCH" THEN 87Ø EL
SE 3ØØ
38Ø ! D
39Ø CALL ZIMMER(A$,Ø,1,1,Ø,"
",1,1)
4ØØ BEF=BEF+1 :: IF A$="HILF
E" THEN CALL HILFE :: GOTO 3
9Ø ELSE 41Ø
41Ø IF A$="O" THEN 162Ø ELSE
 IF A$="S" THEN 51Ø ELSE 39Ø
42Ø ! F
43Ø IF SCHLUESSEL$="DA" THEN
 TE$="GEGENSTAENDE[.........
.............................
........SCHLUESSEL" :: CALL
SPRITE(#1,116,2,65,65)
44Ø CALL ZIMMER(A$,Ø,Ø,1,1,T
E$,3,1)
45Ø TE$=""
46Ø BEF=BEF+1 :: CALL DELSPR
ITE(#1):: IF A$="HILFE" THEN
 CALL HILFE :: GOTO 43Ø ELSE
 47Ø
47Ø IF A$="NIMM SCHL" AND DA
REI<3 AND SCHLUESSEL$="DA" T
HEN CALL SAY("O+K"):: BB$="S
CHLUESSEL" :: DABEI=DABEI+1
:: SCHLUESSEL$="DABEI"
48Ø IF A$="VERL SCHL" AND SC
HLUESSEL$="DABEI" THEN CALL
SAY("O+K"):: DABEI=DABEI-1 :
: BB$="" :: SCHLUESSEL$="DA"
49Ø IF A$="W" THEN 162Ø ELSE
 IF A$="S" THEN 66Ø
5ØØ GOTO 43Ø
51Ø ! G
52Ø CALL SPRITE(#1,1Ø8,2,8,2
8)
53Ø CALL ZIMMER(A$,1,Ø,Ø,Ø,"
VOR IHNEN IST EIN RUNDER....
.............................
KNOPF AN DER WAND]",8,1)
54Ø BEF=BEF+1 :: CALL DELSPR
ITE(#1):: IF A$="HILFE" THEN
 CALL HILFE :: GOTO 52Ø ELSE
 55Ø
55Ø IF A$="NIMM KNOPF" OR A$
="ZIEHE KNOPF" THEN GOTO 59Ø
56Ø IF A$="N" THEN 38Ø ELSE
IF A$="DREHE KNOPF" THEN TUE
RU$="OFFEN" :: GOTO 57Ø ELSE
 52Ø
57Ø CALL DELSPRITE(#1):: FOR
 A=1 TO 2Ø :: SOUN=INT(RND*3
ØØ)+11Ø :: CALL SOUND(-2ØØØ,
SOUN,A,SOUN+19,A):: NEXT A
58Ø GOTO 52Ø
59Ø TEXT$(1)="SIE HABEN EINE
N KURZSCHLUSS.............
.............VERURSACHT, UN
D SIND DURCH.."
6ØØ TEXT$(2)="EINEN STROMSCH
LAG GETOETET.............
.............WORDEN." :: CA
LL ENDE(TEXT$())
61Ø ! H
62Ø CALL ZIMMER(A$,1,1,Ø,Ø,"
",1,1)
63Ø TE$=""
64Ø BEF=BEF+1 :: IF A$="HILF
E" THEN CALL HILFE :: GOTO 6
2Ø ELSE 65Ø
65Ø IF A$="N" THEN 162Ø ELSE
 IF A$="O" THEN 66Ø ELSE 62Ø
66Ø ! I
67Ø CALL ZIMMER(A$,1,1,Ø,1,"
",1,1)
68Ø BEF=BEF+1 :: IF A$="HILF
E" THEN CALL HILFE :: GOTO 6
7Ø ELSE 69Ø
69Ø IF A$="N" THEN 42Ø ELSE
IF A$="O" THEN 182Ø ELSE IF
A$="W" THEN 61Ø ELSE GOTO 67
Ø
7ØØ ! J
71Ø IF KISTE$="DA" THEN TE$=
"GEGENSTAENDE[.............
.............................
...KISTE" :: CALL SPRITE(#1,
96,2,6Ø,6Ø)
```

```
720 CALL ZIMMER(A$,0,1,0,0,T
E$,11,1)
730 TE$=""

740 BEF=BEF+1 :: IF A$="NIMM
 KISTE" AND KISTE$="DA" AND
DABEI<1 THEN CALL SIEG :: CA
LL DELSPRITE(#1):: DABEI=DAB
EI+1 :: CC$="KISTE" :: KISTE
$="DABEI" :: GOTO 710

750 IF A$="O" THEN CALL DELS
PRITE(#1):: GOTO 760 ELSE 71
0
760 ! K
770 CALL ZIMMER(A$,1,1,1,1,T
E$,1,1)
780 TE$=""
790 BEF=BEF+1 :: IF A$="N" T
HEN 2420 ELSE IF A$="O" THEN
 870 ELSE IF A$="W" THEN 840
 ELSE IF A$="S" THEN 960
800 IF A$="OEFFNE TUER" THEN
 810 ELSE 770
810 DISPLAY AT(4,16):"WIE\["
 :: ACCEPT BEEP AT(6,16)SIZE
(14):B$
820 IF B$="BENUTZE SCHL" AND
 SCHLUESSEL$="DABEI" AND KAM
MER$="ZU" THEN 830 ELSE 770
830 CALL SAY("O+K"):: KAMMER
$="OFFEN" :: TE$="DIE TUER I
ST OFFEN]" :: GOTO 770
840 IF A$="W" AND KAMMER$="Z
U" THEN DISPLAY AT(22,1):"DI
E TUER IST ABGESCHLOSSEN]" :
: CALL SAY("SORRY")ELSE 860
850 FOR WAR=1 TO 500 :: NEXT
 WAR :: GOTO 770
860 IF A$="W" AND KAMMER$="O
FFEN" THEN 700
870 ! L
880 CALL ZIMMER(A$,0,0,0,1,"
",0,1)
890 BEF=BEF+1 :: IF A$="W" T
HEN 760 ELSE IF A$="RUNTER"
THEN 290 ELSE 880
900 ! M
910 IF RING$="DABEI" THEN TE
$="DIE WOLKE IST UNGE_.....
.............................
...FAEHRLICH]" ELSE GOTO 940
920 CALL SPRITE(#1,52,2,70,7
0):: CALL SCREEN(5):: CALL Z
IMMER(A$,0,1,1,0,TE$,3,1)
930 TE$="" :: BEF=BEF+1 :: C
ALL DELSPRITE(ALL):: IF A$="
S" THEN 1080 ELSE IF A$="O"
THEN 960 ELSE 910
940 TEXT$(1)="SIE SIND VON D
ER WOLKE VER-.............
.............NICHTET WORDEN
```

LISTINGS

```
, DA SIE ETWAS"
950 TEXT$(2)="WICHTIGES NICH
T BEACHTET.................
.............HABEN." :: CAL
L ENDE(TEXT$())
960 ! N
970 CALL ZIMMER(A$,1,0,0,1,"
",1,1)
980 TE$=""
990 BEF=BEF+1 :: IF A$="N" T
HEN 760 ELSE IF A$="W" THEN
900 ELSE 970
1000 ! O
1010 CALL ZIMMER(A$,0,1,1,0,
"",1,1)
1020 DISPLAY AT(17,18):"DABE
I["
1030 DISPLAY AT(19,18):AA$ :
: DISPLAY AT(20,18):BB$ :: D
ISPLAY AT(21,18):CC$ :: DISP
LAY AT(22,18):DD$ :: DISPLAY
 AT(23,18):EE$
1040 TE$="" :: DISPLAY AT(24
,18):FF$
1050 IF A$="WARTE" THEN 1060
 ELSE 1070
1060 FOR WAR=1 TO 500 :: NEX
T WAR
1070 BEF=BEF+1 :: IF A$="O"
THEN 2170 ELSE IF A$="S" THE
N 1330 ELSE 1010
1080 ! P
1090 IF KOBOLD$="DA" THEN TE
$="EIN KOBOLD STEHT VOR IHNE
N]" :: CALL SPRITE(#1,140,2,
75,75)
1100 CALL ZIMMER(A$,1,0,1,0,
TE$,9,1)
1110 TE$=""
1120 BEF=BEF+1 :: IF KOBOLD$
="WEG" THEN 1190 ELSE IF A$<
>"TOETE KOBOLD" THEN GOTO 12
00
1130 IF A$="TOETE KOBOLD" AN
D GIFT$="DA" THEN GOTO 1220
1140 IF A$="TOETE KOBOLD" AN
D GIFT$="DABEI" THEN 1150 EL
SE 1190
1150 DISPLAY AT(4,16):"WIE\[
" :: ACCEPT BEEP AT(6,16)SIZ
E(15)VALIDATE(UALPHA):A$
1160 IF A$="BENUTZE GIFT" TH
EN KOBOLD$="WEG" ELSE GOTO 1
200
1170 CALL DELSPRITE(#1):: FO
R FREI=18 TO 24 :: CALL HCHA
R(FREI,1,32,32):: NEXT FREI
1180 DISPLAY AT(21,1):"DER K
OBOLD HAT SICH.............
.....................IN LU
FT AUFGELOEST]" :: FOR WAR=1
```

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE

TI REVUE
Das Magazin für TI 99-4A

KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre TI REVUE nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Ihr" Kiosk nicht beliefert wurde? Kein Problem! Für ganze 60 DM liefern wir per Post zwölf Hefte ins Haus (Ausland 80 DM). Einfach den Bestellschein auf der nächsten Seite ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). TI REVUE kommt dann pünktlich ins Haus.

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen– Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für zwölf Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

DAS ANGEBOT: KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen ab sofort die TI-REVUE: KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!

Wir veröffentlichen nur Kleinanzeigen privater Inserenten, keine gewerblichen Anzeigen. Die kosten pro Millimeter DM 5,– plus Mehrwertsteuer!

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!

Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!

Die Insertion ist nicht vom Kauf des Heftes abhängig!

RESERVIERUNGS-SERVICE

Selbstverständlich denken wir bei diesem Special-Heft auch wieder an jene unter unseren Lesern, die keine Zeit haben, diese ganzen Listings einzugeben. Für sie hält der Kassettenservice das gesamte Angebot dieses Heftes auf vier Disketten bereit. Diese Disketten sind nur im Paket zu beziehen, sie kosten geschlossen 75,– DM. Einfach den Coupon auf der nächsten Seite ausschneiden, ausfüllen und absenden. Diese Seite ist so gestaltet, daß keine Information des Heftes verloren geht.

Natürlich berücksichtigen wir auch jene TI-User, die noch kein Disketten-Laufwerk besitzen. Für Sie haben wir alle Listings dieses Heftes auf Kassetten zusammengefaßt. Das Programmpaket kostet komplett DM 30,–. Es kann nur geschlossen bezogen werden.

Benutzen Sie bitte auch den Bestellcoupon auf der nächsten Seite.

Ein wichtiger Hinweis: Wir versenden nicht per Nachnahme oder Vorausrechnung!

Sie können aber auch den Betrag auf eines unserer Konten überweisen. Hier die Kontonommern:
Postgiro – Amt München, Konto W. Seibt / TI-Verlag, Kto-Nr. 416 103-805, BLZ 700 100 80 oder
Raiffeisenbank Lohhof, Kto.-Nr. 306 959, BLZ 701 695 85.

Bitte Absender (komplett) deutlich schreiben – am besten in Druckbuchstaben! Und Hinweis, was gewünscht wird, nicht vergessen! Z.B. Disketten TI-Special Nr. 4!

ABO SERVICE-KARTE

TI/Sp.

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

TI REVUE

Abo-Service Sp
Postfach 1107
8044 UNTERSCHLEISSHEIM

Coupon

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten zwölf Ausgaben an untenstehende Anschrift. Sollte ich nicht vier Wochen vor Ablauf schriftlich kündigen, läuft diese Abmachung automatisch weiter.

Name _____

Vorname _____

Straße/Hausnr. _____

Plz/Ort _____

Ich bezahle:

☐ per beiliegendem Verrechnungsscheck
☐ gegen Rechnung
☐ bargeldlos per Bankeinzug von meinem Konto

bei (Bank) und Ort _____

Kontonummer _____

Bankleitzahl _____

(steht auf jedem Kontoauszug)

Unterschrift _____

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift _____

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen

die Listings dieses Heftes auf

☐ 4 Disketten zum Preis von DM 75,–
☐ Die Kassetten zum Preis von DM 30,–

Ich zahle:

Bar – per beigefügtem Geld ( )
per beigefügtem Scheck ( )
Gegen Bankabbuchung am Versandtag ( )

Zutreffendes bitte ankreuzen!

Meine Bank (mit Ortsname) .....

Meine Kontonummer .....

Meine Bankleitzahl ..... **(steht auf jedem Bankauszug)**

Vorname ..... **Nachname** .....

Str./Nr. ..... **PLZ / Ort** .....

Hiermit bestätige ich mit meiner Unterschrift, Ihre Verkaufsbedingungen gelesen zu haben und zu akzeptieren.

Unterschrift .....

Bitte ausschneiden und einsenden an
TI-REVUE
Special-Service 4
Postfach 1107
8044 Unterschleißheim

special ASSEMBLER special

Über 90 Seiten Alles über Assembler für den TI 99/4A

MACHEN SIE MEHR AUS IHREM TI MIT ASSEMBLER

IM BAHNHOFSBUCH-HANDEL ODER BEIM VERLAG!

ANZEIGENSERVICE

Die große Börse für jeden Zweck in der TI REVUE. Kostenlos für Privat-Inserenten. Spottbillig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Achtung! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt nicht TI-typisch ist oder gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatanbieter: maximal acht Zeilen à 28 Anschläge. Für gewerbliche Anbieter: 5 DM p. mm.

TI
Anzeigenabteilg.
Postfach 1107
8044 Lohhof

Name _____
Vorname _____
Straße/Hausnr. _____
PLZ/Ort _____

TI REVUE
Das Magazin für TI 99-4A

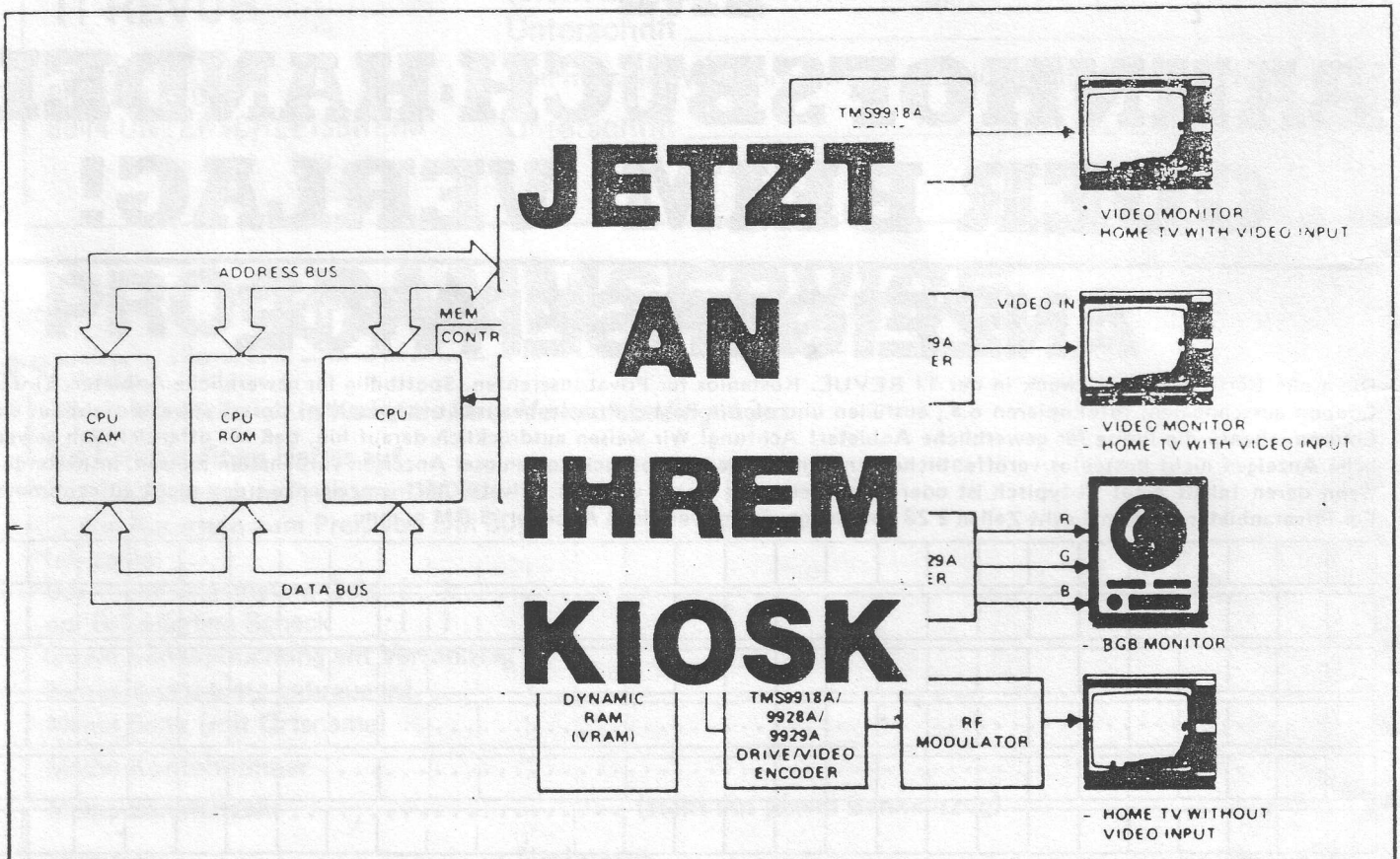
Nr. 4/86/April -DM 5,50 / ÖS 46 / SRF 5,50

Multiplan für den TI 99/4A

Steuern über den I/O-Port

Deutsche Tastatur für Ihren TI

Assembler-leicht gemacht



FIGUR 1-1 – SYSTEM BLOCK DIAGRAMM

Type	Ausgang	Farbnorm
TMS9918A	Composite Video Signal	NTSC
TMS9928A	Farb Differenz Signale	NTSC
TMS9929A	Farb Differenz Signale	PAL/SECAM

```
 TO 500 :: NEXT WAR
1190 IF A$="N" THEN 900 ELSE
 IF A$="S" THEN 1460 ELSE 10
90
1200 TEXT$(1)="SIE SIND VOM
KOBOLD VERZAU-.............
.............BERT WORDEN,
UND KOENNEN.."
1210 TEXT$(2)="NICHT WEITERS
UCHEN." :: CALL ENDE(TEXT$()
)
1220 TEXT$(1)="SIE HABEN EIN
E WHCHTIGE.................
.............SACHE VERGESS
EN, SO DASS DER"
1230 TEXT$(2)="KOBOLD SIE VE
RZAUBERN KONNTE" :: CALL END
E(TEXT$())
1240 ! Q
1250 IF KISTE$="DABEI" THEN
TE$="DURCH DAS GEWICHT DER K
ISTE.......................
.....HAT SICH DICHT NEBEN IH
NEN.......................
.....EINE FALLTUER GEOEFFNET
]"
1260 IF KISTE$="DABEI" THEN
CO1=0 ELSE CO1=1
1270 CALL ZIMMER(A$,0,1,0,0,
"",CO1,1)
1280 TE$=""
1290 BEF=BEF+1 :: IF A$="RUN
TER" AND KISTE$="DABEI" THEN
 CALL CHAR(33,"37021AB4891CD
E5F")ELSE 1320
1300 IF DRU$="J" THEN CALL C
HAR(35,"1234567876543210")
1310 RUN "DSK1.PORTAL"
1320 IF A$="O" THEN 1330 ELS
E 1250
1330 ! R
1340 IF SCHLANGE$="DA" THEN
TE$="EINE SCHLANGE BEDROHT S
IE]" :: CALL SPRITE(#1,56,2,
70,65)
1350 CALL ZIMMER(A$,1,0,0,1,
TE$,9,1)
1360 TE$=""
1370 BEF=BEF+1 :: CALL DELSP
RITE(ALL):: IF A$<>"TOETE SC
HL" AND SCHLANGE$="DA" THEN
GOTO 1440
1380 IF A$="TOETE SCHL" AND
SCHLANGE$="DA" THEN 1390 ELS
E 1430
1390 CALL SPRITE(#1,56,9,70,
65):: DISPLAY AT(4,16):"WIE\
[" :: ACCEPT BEEP AT(6,16)SI
ZE(15)VALIDATE(UALPHA):A$
1400 IF A$="WIRF STEIN" AND
STEIN$="DABEI" THEN SCHLANGE
$="WEG" :: STEIN$="" :: DABE
I=DABEI-1 :: FF$=""
1410 DISPLAY AT(22,1):"SIE H
ABEN DIE.....................
.....................SCHLA
NGE GETOETET]...."
1420 CALL DELSPRITE(#1):: FO
R WAR=1 TO 500 :: NEXT WAR
1430 IF A$="N" THEN 1000 ELS
E IF A$="W" THEN 1240 ELSE 1
340
1440 TEXT$(1)="SIE SIND VON
DER SCHLANGE...............
.............GEBISSEN WORD
EN, UND SIND AN"
1450 TEXT$(2)="DEN FOLGEN GE
STORBDN." :: CALL ENDE(TEXT$
())
1460 ! S
1470 CALL ZIMMER(A$,1,1,1,0,
"",1,1)
1480 BEF=BEF+1 :: IF A$="N"
THEN 1080 ELSE IF A$="S" THE
N 2420 ELSE IF A$="O" THEN 1
490 ELSE 1470
1490 ! T
1500 CALL ZIMMER(A$,0,1,1,1,
"",1,1)
1510 !
1520 BEF=BEF+1 :: IF A$="O"
THEN 1530 ELSE IF A$="S" THE
N 2420 ELSE IF A$="W" THEN 1
460 ELSE 1500
1530 ! U
1540 IF MAUER$<>"WEG" THEN T
E$="SIE STEHEN VOR EINER MAU
ER]" :: CO=100
1550 CALL ZIMMER(A$,0,1,0,1,
TE$,CO,1)
1560 TE$=""
1570 BEF=BEF+1 :: IF A$="WIR
F DYNAMIT" AND DYNAMIT$="DAB
EI" THEN 1580 ELSE 1610
1580 DD$="" :: DABEI=DABEI-1
 :: CALL CHAR(134,"526735514
89F5167"):: CALL SOUND(900,-
7,0,110,15):: MAUER$="WEG" :
: DYNAMIT$="WEG" :: CALL CHA
R(134,"00000000000000000")
1590 FOR A=18 TO 24 :: CALL
HCHAR(A,1,32,32):: NEXT A ::
 DISPLAY AT(22,1):"DIE MAUER
 IST.......................
.....................ZERSTOERT
]" :: CALL SAY("#GOOD WORK")
1600 FOR WAR=1 TO 500 :: NEX
T WAR
1610 IF A$="O" THEN 2300 ELS
E IF A$="W" AND MAUER$="WEG"
 THEN 1490 ELSE 1540
```

```
1620 ! E
1630 IF POI=0 THEN TE$="SIE
BEFINDEN SICH.............
.......................IN E
INEM RAUM DES.............
.......................SCHL
OSSES]" :: POI=1
1640 CALL ZIMMER(A$,1,1,1,1,
TE$,1,1)
1650 TE$="" :: DISPLAY AT(19
,18):AA$ :: DISPLAY AT(20,18
):BB$ :: DISPLAY AT(21,18):C
C$ :: DISPLAY AT(22,18):DD$
:: DISPLAY AT(23,18):EE$
1660 DISPLAY AT(17,18):"DABE
I[" :: DISPLAY AT(24,17):FF$
1670 IF A$<>"WARTE" THEN 170
0
1680 FOR WAR=1 TO 600 :: NEX
T WAR
1690 !
1700 BEF=BEF+1 :: IF A$="HIL
FE" THEN CALL HILFE :: GOTO
1630 ELSE 1710
1710 IF A$="N" THEN 250 ELSE
 IF A$="S" THEN 610 ELSE IF
A$="W" THEN 380 ELSE IF A$="
O" THEN 420
1720 IF A$="OEFFNE TUER" THE
N DISPLAY AT(10,17)SIZE(15):
"TUEREN" :: DISPLAY AT(11,18
)SIZE(10):"SIND" :: DISPLAY
AT(12,17)SIZE(15):"OFFEN"
1730 GOTO 1630
1740 ! Z
1750 IF SPEER$="DA" THEN TE$
="GEGENSTAENDE[.............
.............................
....SPEER" :: CALL SPRITE(#1
,112,2,70,70)
1760 CALL ZIMMER(A$,0,0,1,0,
TE$,16,1)
1770 TE$=""
1780 BEF=BEF+1 :: CALL DELSP
RITE(#1):: IF A$="HILFE" THE
N CALL HILFE :: GOTO 1750 EL
SE 1790
1790 IF A$="NIMM SPEER" AND
DABEI<3 AND SPEER$="DA" THEN
 CALL SAY("O+K"):: CC$="SPEE
R" :: DABEI=DABEI+1 :: SPEER
$="DABEI"
1800 IF A$="VERL SPEER" AND
SPEER$="DABEI" THEN CALL SAY
("O+K"):: DABEI=DABEI-1 :: S
PEER$="DA" :: CC$=""
1810 IF A$="S" THEN 1820 ELS
E 1750
1820 ! . .
1830 !  A
```

LISTINGS

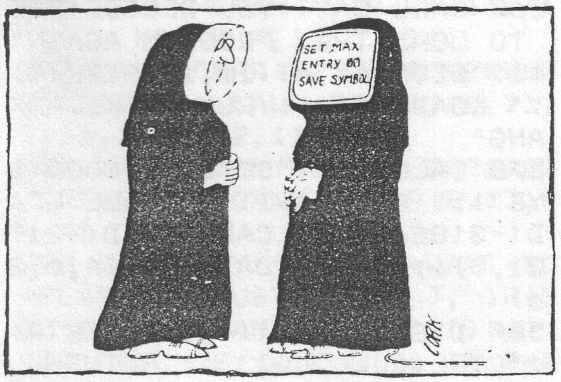
```
1840 IF WACHE$="DA" THEN TE$
="EINE WACHE BEWACHT DIE....
.............................
..OST_TUER]" :: CALL SPRITE(
#1,120,2,56,110)
1850 CALL ZIMMER(A$,1,1,0,1,
TE$,5,1)
1860 TE$=""
1870 BEF=BEF+1 :: CALL DELSP
RITE(#1):: IF A$="HILFE" THE
N CALL HILFE :: GOTO 1840 EL
SE 1880
1880 IF A$="VERSCH WACHE" TH
EN GOSUB 2440 :: GOTO 1840
1890 !
1900 IF A$="N" THEN 1740 ELS
E IF A$="W" THEN 660 ELSE 19
10
1910 IF A$="TOETE WACHE" THE
N 1920 ELSE 1950
1920 TEXT$(1)="NACH EINEM HA
RTEN KAMPF WUR-.............
.............DEN SIE VON D
ER WACHE "
1930 TEXT$(2)="BESIEGT. SIE
WURDEN VERNICH-.............
.............TET."
1940 CALL ENDE(TEXT$())
1950 IF A$="O" AND WACHE$="W
EG" THEN 1980 ELSE IF A$="O"
 AND WACHE$="DA" THEN 1960 E
LSE 1840
1960 TEXT$(1)="SIE WURDEN VO
N DER WACHE BE-.............
.............MERKT UND GET
OETET..."
1970 TEXT$(2)="" :: CALL END
E(TEXT$())
1980 ! . .
1990 !  O
2000 CALL ZIMMER(A$,1,0,0,1,
"",1,0)
2010 BEF=BEF+1 :: IF A$="HIL
FE" THEN CALL HILFE :: GOTO
2000 ELSE 2020
2020 IF A$="N" THEN 2030 ELS
E IF A$="W" THEN 1820 ELSE I
F A$="HOCH" THEN 2330 ELSE 2
000
2030 TEXT$(1)="HINTER DIESER
 TUER WAR EINE.............
.............FALLGRUBE.BEI
 DEM STURZ SIND"
2040 TEXT$(2)="SIE UMS LEBEN
 GEKOMMEN."
2050 CALL ENDE(TEXT$())
2060 ! Y
2070 IF DYNAMIT$="DA" THEN T
E$="GEGENSTAENDE[...........
.............................
.......DYNAMIT" :: CALL SPRI
```

```
TE(#1,124,2,55,56)
2080 CALL ZIMMER(A$,0,0,1,0,
TE$,9,1)
2090 TE$=""
2100 BEF=BEF+1 :: CALL DELSP
RITE(#1):: IF A$="S" THEN 21
70 ELSE 2110
2110 IF A$="NIMM DYNAMIT" AN
D DABEI<2 AND DYNAMIT$="DA"
THEN CALL SAY("O+K"):: DD$="
DYNAMIT" :: DABEI$(I)="DYNAM
IT" :: DABEI=DABEI+1 :: CALL
 DELSPRITE(#1):: DYNAMIT$="D
ABEI"
2120 IF A$="VERL DYNAMIT" TH
EN 2140
2130 GOTO 2070
2140 TEXT$(1)="DAS DYNAMIT I
ST DURCH DEN.....
.....AUFSCHLAG EXP
LODIERT. DIE "
2150 TEXT$(2)="SIND GETOETET
 WORDEN."
2160 CALL ENDE(TEXT$())
2170 ! X
2180 CALL ZIMMER(A$,1,1,1,1,
"",1,1)
2190 TE$=""
2200 BEF=BEF+1 :: IF A$="N"
THEN 2060 ELSE IF A$="O" THE
N 2420 ELSE IF A$="S" THEN 2
220 ELSE IF A$="W" THEN 1000
2210 GOTO 2180
2220 ! W
2230 IF RING$="DA" THEN TE$=
"GEGENSTAENDE[.....
.....
...RING" :: CALL SPRITE(#1,1
20,2,90,90)
2240 CALL ZIMMER(A$,1,1,1,0,
TE$,14,1)
2250 TE$=""
2260 BEF=BEF+1 :: CALL DELSP
RITE(#1):: IF A$="N" THEN 21
70 ELSE IF A$="S" THEN 2300
ELSE IF A$="O" THEN 2330
2270 IF A$="NIMM RING" AND D
ABEI<2 AND RING$="DA" THEN C
ALL SAY("O+K"):: DAREI=DABEI
+1 :: EE$="RING" :: RING$="D
ABEI"
2280 IF A$="VERL RING" AND R
ING$="DABEI" THEN CALL SAY("
O+K"):: DABEI=DABEI-1 :: RIN
G$="DA" :: EE$=""
2290 GOTO 2230
2300 ! V
2310 CALL ZIMMER(A$,1,0,1,1,
"",1,1)
2320 BEF=BEF+1 :: IF A$="S"
THEN 2420 ELSE IF A$="N" THE
N 2220 ELSE IF A$="W" THEN 1
530 ELSE 2310
2330 ! . .
2340 !  U
2350 CALL CHAR(120,"00000000
0000000102040404040201000000
0000F060F0080402020202040 8F0
")
2360 !
2370 IF TUERU$="OFFEN" THEN
D=1 ELSE D=0
2380 !
2390 CALL ZIMMER(A$,0,0,0,D,
"",0,1)
2400 BEF=BEF+1 :: IF A$="RUN
TER" THEN 1980 ELSE 2410
2410 IF A$="W" AND TUERU$="O
FFEN" THEN 2220 ELSE 2350
2420 TEXT$(1)="DIESE TUER WA
R EINE FALLE. .....
.....SIE SIND IN D
EN SCHLOSS-..."
2430 TEXT$(2)="GRABEN GEFALL
EN UND ERTRUN-.....
.....KEN." :: CALL
 ENDE)TEXT$())
2440 ! WACHE
2450 IF WACHE$="WEG" THEN RE
TURN ELSE CALL SPRITE(#1,120
,5,56,110):: DISPLAY AT(4,16
):"WIE\[" :: ACCEPT BEEP AT(
6,16)SIZE(15)VALIDATE(UALPHA
):A$
2460 IF A$="WIRF SPEER" AND
SPEER$="DABEI" THEN WACHE$="
WEG" :: CC$="" :: DABEI=DABE
I-1 :: CALL SAY("#NICE TRY")
:: CALL DELSPRITE(#1):: GOSU
B 2480 ELSE RETURN
2470 RETURN
2480 FOR A=18 TO 24 :: CALL
HCHAR(A,1,32,32):: NEXT A ::
 DISPLAY AT(20,1):"WACHE IST
 WEG]" :: FOR A=1 TO 500 ::
NEXT A :: RETURN
2490 ! HILFE
2500 SUB HILFE
2510 CALL CLEAR :: CALL SCRE
EN(16)
2520 DISPLAY AT(2,6):"B E F
E H L E [" 
2530 DISPLAY AT(5,1):"NIMM^
VERL_IERE_^ TOETE^.....
.....DRUECK
E^ DREHE^ ZIEHE^.....
....."
2540 DISPLAY AT(9,1):"WIRF^
VERSCH_EUCHE_^ TRINKE.....
....."
2550 DISPLAY AT(12,1):"WEITE
```

```
RE ABKUERZUNGEN:"
2560 DISPLAY AT(14,1):"SCHL_
UESSEL^ SCHL_ANGE "
2570 CALL SAY("PRESS J OR N"
)
2580 DISPLAY AT(24,1):"WEITE
R SPIELEN\ J ODER N"
2590 CALL KEY(0,K,S):: IF S=
0 THEN 2590
2600 IF K=110 OR K=78 THEN C
ALL INIT :: CALL LOAD(-31806
,0):: CALL LOAD(-31931,0)::
CALL LOAD(-31804,0,36):: GOT
O 2600
2610 CALL SCREEN(2):: CALL C
LEAR :: SUBEND
2620 !
2630 !
2640 !
2650 ! ************
2660 ! ***      ***
2670 ! *** SIEG ***
2680 ! ***      ***
2690 ! ************
2700 !
2710 !
2720 SUB SIEG
2730 FOR A=18 TO 24 :: CALL
HCHAR(A,1,32,32):: NEXT A
2740 DISPLAY AT(22,1):"BRAVO
]]] SIE.....
.....HABEN
 DEN SCHATZ GEFUNDEN ]"
2750 FOR A=1 TO 3 :: CALL SO
UND(400,220,0,175,10):: FOR
I=1 TO 100 :: NEXT I :: CALL
 SOUND(400,175,0,349,10)
2760 FOR I=1 TO 100 :: NEXT
I :: CALL SOUND(400,196,0,13
1,10)
2770 FOR I=1 TO 100 :: NEXT
I :: CALL SOUND(700,131,0,26
2,10)
2780 FOR I=1 TO 300 :: NEXT
I :: CALL SOUND(400,131,0,26
2,10)
2790 FOR I=1 TO 100 :: NEXT
I :: CALL SOUND(400,196,0,13
1,10)
2800 FOR I=1 TO 100 :: NEXT
I :: CALL SOUND(400,220,0,17
5,10)
2810 FOR I=1 TO 100 :: NEXT
I :: CALL SOUND(700,175,0,34
9,10)
2820 FOR I=1 TO 400 :: NEXT
I
2830 NEXT A
2840 DISPLAY AT(22,1):"ES IS
T DREI UHR^ ZEIT DAS.....
.....SCHLO
SS ZU VERLASSEN]"
2850 FOR WAR=1 TO 500 :: NEX
T WAR :: SUBEND
2860 SUB DRUCK
2870 OPEN #1:"PIO"
2880 PRINT #1:CHR$(14);"UNTE
RGESCHOSS"
2890 PRINT #1:CHR$(10)
2900 PRINT #1:CHR$(27);"3";C
HR$(20);
2910 READ J
2920 IF J=0 THEN 2950
2930 PRINT #1:CHR$(J);
2940 GOTO 2910
2950 CLOSE #1 :: SUBEND
2960 SUB ZIMMER(A$,A,D,B,C,T
E$,CO1,CO2)
2970 CALL CLEAR :: CALL SCRE
EN(2):: FOR I=1 TO 14 :: CAL
L COLOR(I,2,2):: NEXT I
2980 PRINT TAB(2);"333333333
3333"
2990 FOR I=1 TO 13 :: PRINT
TAB(1);"1.....0" ::
NEXT I
3000 PRINT TAB(2);"222222222
2222"
3010 PRINT : : : : : : : :
3020 IF A=1 THEN DISPLAY AT(
1,7)SIZE(3):"*+,"
3030 IF B=1 THEN DISPLAY AT(
15,7)SIZE(3):"'()"
3040 IF C=1 THEN CALL HCHAR(
7,3,36):: CALL HCHAR(8,3,37)
:: CALL HCHAR(9,3,38)
3050 IF D=1 THEN CALL HCHAR(
7,17,45):: CALL HCHAR(8,17,4
6):: CALL HCHAR(9,17,47)
3060 FOR I=1 TO 2 :: CALL CO
LOR(I,9,1):: NEXT I :: CALL
COLOR(3,15,1)
3070 FOR I=4 TO 13 :: CALL C
OLOR(I,4,1):: NEXT I
3080 CALL COLOR(14,11,1,0,4,
1,13,8,1)
3090 IF TE$="" THEN 3100 ELS
E DISPLAY AT(18,1):TE$
3100 IF CO2=0 THEN CALL HCHA
R(8,10,136):: CALL HCHAR(9,1
0,137):: CALL HCHAR(8,11,138
):: CALL HCHAR(9,11,139)ELSE
 CALL COLOR(#2,CO2)
3110 IF CO1=0 THEN CALL HCHA
R(8,10,130):: CALL HCHAR(9,1
0,131):: CALL HCHAR(8,11,132
):: CALL HCHAR(9,11,133):: G
OTO 3130
3120 IF CO1=100 THEN CALL VC
HAR(2,10,134,13)ELSE CALL CO
LOR(#1,CO1)
```

```
3130 DISPLAY AT(4,16):"BEFEH
L\[" :: ACCEPT BEEP AT(6,16)
SIZE(16)VALIDATE(UALPHA):A$
3140 SUBEND
3150 SUB ENDE(TEXT$())
3160 CALL CLEAR :: CALL CHAR
SET :: ON BREAK NEXT :: CALL
 MAGNIFY(3):: CALL SCREEN(9)
3170 FOR A=0 TO 8 :: CALL CO
LOR(A,16,1):: NEXT A
3180 CALL CHAR(120,"00C7CF3F
1B1F1E0F06040F1763600000000C6
E6F8B0F0F0E0C040E0B08C0C")
3190 DISPLAY AT(5,1):TEXT$(1
);: DISPLAY AT(9,1):TEXT$(2)
3200 CALL SPRITE(#1,120,2,90
,125)
3210 DISPLAY AT(16,1):"UM NO
CH EINMAL ZU SPIELEN,.......
.........................BITTE
 'J' EINGEBEN, SONST.........
.........................'N'."
3220 DISPLAY AT(24,1):"IHRE
WAHL?:" :: ACCEPT BEEP AT(24
,13):A$
3230 IF A$<>"J" THEN GOTO 32
80

3240 CALL DELSPRITE(#1):: CA
LL CLEAR
3250 DISPLAY AT(12,1):"BITTE
 GEDULDEN SIE SICH EINE.....
.........................KURZE
 ZEIT. DAS PROGRAMM"
3260 DISPLAY AT(16,1):"WIRD
WIEDER EINGELESEN !"
3270 RUN "DSK1.ANFANG"
3280 CALL DELSPRITE(ALL):: C
ALL CLEAR
3290 DISPLAY AT(12,1):" DAS
PROGRAMM..WIRD AUS DEM......
.........................SPE
ICHER ENTFERNT !!! "
3300 CALL SAY("GOODBYE"):: F
OR A=1 TO 700 :: NEXT A
3310 CALL INIT :: CALL LOAD(
-31806,0):: CALL LOAD(-31931
,0):: CALL LOAD(-31804,0,36)
3320 SUBEND
```



LISTINGS

```
10 ! **********************
11 ! *   ZAUBERSCHLOSS    *
12 ! * (Teil 4: "PORTAL") *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *   Thorsten Ziehm   *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *      32K-Erw.      *
23 ! *    Speech-Synth.   *
24 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     4322 Bytes     *
28 ! *                    *
29 ! **********************
100 CALL CHARPAT(33,C$):: IF
 C$="37021AB4891CDE5F" THEN
140
110 S$="GAMES" :: CALL SPGET
(S$,SS$):: SSS$=SEG$(SS$,1,7
0):: CALL SAY("BE+FOR+YOU+CA
N+SEE+THE1+LAST+SCREEN,YOU+M
UST+LOAD+AND+PLAY+THE1")
120 CALL SAY(,SSS$):: CALL S
AY("WITH+THE1+NAME LOAD,WHIC
H+IS+ON+THIS+DISKETTE,TOO")
130 GOTO 570
140 CALL CLEAR
150 CALL SCREEN(2):: CALL CH
ARSET
160 DISPLAY AT(6,1):"SIE SIN
D DURCH DIE FALLTUER........
.........................IN EINE
 DUNKLE OEFFNUNG GE-"
170 DISPLAY AT(10,1):"STIEGE
N. DORT BEFAND SICH.........
.........................EINE R
UTSCHE, DIE SIE DIREKT"
180 DISPLAY AT(14,1):"ZUM HA
UPTEINGANG GEBRACHT.........
.........................HAT."
190 DISPLAY AT(22,1):"IHRE M
ISSION IST HIERMIT BE-......
.........................ENDET.
 DER SCHATZ IST IHRER."
200 CALL SCREEN(16)
210 CALL SAY("#GOOD WORK")
220 FOR SD=1 TO 2500 :: NEXT
 SD
230 CALL CHARSET :: CALL CLE
AR :: CALL SCREEN(2)
240 CALL CHAR(140,"444444444
4444444",129,"444444FFFF4444
44")
250 CALL CHAR(130,"4444EEEE4
4440000",131,"C44464544C4446
45",132,"C44464FFFF444645",1
```

```
33,"C444EEFE4C44Ø2Ø1",134,"4
546EEEE54644Ø8Ø")
26Ø CALL CHAR(135,"4546444C5
46444C4",136,"454644FFFF6444
C4",137,"FFC5C5C5C5FFC5C5",1
38,"C5C5C5C5C5CDC5C5",139,"C
5C5C5FFFFC5C5FF")
27Ø !
28Ø CALL COLOR(8,11,1,13,11,
1)
29Ø CALL COLOR(14,11,1,2,16,
1,3,16,1,4,16,1,5,3,1,6,3,1,
7,3,1,8,3,1,9,3,1,1Ø,3,1,11,
3,1)
3ØØ A$(1)="a a a a.........
...a a a a aaaaaaa.........
...aaaaaaa aaaaaaa.........
...aaaaaaa aaaaaaa.........
...aaaaaaa "
31Ø A$(2)="aaaaaaa.........
...aaaaaaa aaaaaaa.........
...aaaaaaa aaaaaaa a a a a a
 a aaaaaaa aaaaaaaaaaaaaaaaa
aaaaaaaaaa "
32Ø A$(3)="aaaaaaaaaaaaaaaaa
aaaaaaaaaa aaa^_aaaaaaaaaaaa
aaaaa^_aaa aaagfaaaaaaaaaaaa
aaaaagfaaa aaadeaaaaaaaaaaaa
aaaaadeaaa "
33Ø A$(4)="aaaaaaaaaaaaaaaaa
aaaaaaaaaa aaaaaaaaaaaaaaaaa
aaaaaaaaaa aaaaaaaaaccccccccc
caaaaaaaaa aaaaaaaaabj.....k
baaaaaaaaa "
34Ø A$(5)="aaaaaaaaab j...k
baaaaaaaaa aaaaaaaaab..j k..
baaaaaaaaa aaaaaaaaab...u...
baaaaaaaaa aaaaaaaaab...v...
baaaaaaaaa "
35Ø A$(6)="aaaaaaaaab...w...
baaaaaaaaa aaaaaaaaab..k j..
baaaaaaaaa aaaaaaaaab k...j
baaaaaaaaa aaaaaaaaabk.....j
baaaaaaaaa "
36Ø CALL CHAR(42,"ØØØØØØØØØØ
ØØØØØØ",43,"FFFFFFFFFFFFFFFF
",47,"FF818181818181FF",97,"
FDFDFDØØDFDFDFØØ",98,"5A5A5A
5A5A5A5A5A")
37Ø CALL CHAR(99,"FFØØDD5577
ØØFFFF",1ØØ,"FFC1C1C1C1C1FFF
F",1Ø1,"FF8383838383FFFF",1Ø
2,"FF8383838383FF",1Ø3,"FF
C1C1C1C1C1C1FF")
38Ø CALL CHAR(94,"FFF9F1E1C1
C1C1FF",95,"FF9F8F87838383FF
",1Ø6,"8Ø4Ø2Ø1ØØ8Ø4Ø2Ø1",1Ø7
,"Ø1Ø2Ø4Ø81Ø2Ø4Ø8Ø",1Ø8,"FFC
ØA9Ø88848281")
39Ø CALL CHAR(1Ø9,"FFØ3Ø5Ø91
1214181",11Ø,"818284889ØAØCØ
FF",111,"8141211109Ø503FF",1
12,"FFØØØØØØØØØØØØØØ",113,"Ø
ØØØØØØØØØØØØFF")
4ØØ CALL CHAR(114,"8Ø8Ø8Ø8Ø8
Ø8Ø8Ø8Ø",115,"Ø1Ø1Ø1Ø1Ø1Ø1Ø1
Ø1",117,"FF818181FF818181",1
18,"81818181D818581",119,"8
1818181818181FF")
41Ø CALL CHAR(12Ø,"818181818
18181FF")
42Ø DISPLAY AT(1,1):A$(1);A$
(2);A$(3);A$(4);A$(5);A$(6)
43Ø CALL MAGNIFY(3):: CALL C
HAR(81,"Ø2Ø2Ø2Ø2Ø2Ø2Ø2Ø2")
44Ø CALL COLOR(1Ø,11,1,11,1,1
,1,12,11,1)
45Ø CALL CHAR(12Ø,"ØØFØFFFFØ
FØØØØFØFFFFØFØØØØØØØØØØ23EF
EFEC2Ø2Ø23EFEFEC2Ø2Ø2Ø2Ø2Ø2"
)
46Ø CALL SPRITE(#1,12Ø,1Ø,18
,121)
47Ø CALL VCHAR(3,17,81,4)::
FOR SD=1 TO 1ØØØ :: NEXT SD
48Ø FOR TY=16 TO 24 STEP 1 :
: CALL SOUND(-1ØØ,-4,Ø):: CA
LL HCHAR(TY,13,32,7):: ON TY
-15 GOSUB 58Ø,59Ø,6ØØ,61Ø,62
Ø,63Ø,64Ø,65Ø,66Ø :: NEXT TY
49Ø CALL HCHAR(2,1Ø,32,13)
5ØØ CALL MOTION(#1,2,Ø):: FO
R SD=1 TO 9ØØ :: NEXT SD
51Ø CALL DELSPRITE(#1)
52Ø DISPLAY AT(12,1Ø)SIZE(9)
:" E N D E " :: DISPLAY AT(1
1,1Ø)SIZE(9):"........." ::
DISPLAY AT(13,1Ø)SIZE(9):".
........"
53Ø CALL SPGET("GAMES",B$)::
 C$=SEG$(B$,1,7Ø):: CALL SAY
(,C$,"OVER")
54Ø CALL SAY("IF+YOU+WANT+TO
+PLAY+THIS+",,,C$,"AGAIN YOU
+MUST+PRESS,J ,IF+YOU+DO+NOT
+WANT, YOU+MUST+PRESS,N")
55Ø CALL KEY(Ø,K,S):: IF S=Ø
 THEN 55Ø ELSE IF K=74 OR K=
1Ø6 THEN 56Ø ELSE 57Ø
56Ø CALL SAY("YOU DECIDE+DID
 TO LOAD THE1 PROGRAM AGAIN"
):: DISPLAY AT(12,1Ø)SIZE(9)
:" LOADING" :: RUN "DSK1.ANF
ANG"
57Ø CALL SAY("GOODBYE,GOOD+B
YE"):: CALL INIT :: CALL LOA
D(-31Ø8Ø6,Ø):: CALL LOAD(-319
31,Ø):: CALL LOAD(-31Ø84,Ø,3
6)
58Ø DISPLAY AT(16,11)SIZE(7)
:"CTRL(CLLLLLG)" :: RETURN
```

```
590 DISPLAY AT(17,11)SIZE(7)
:"CTRL{LCLLLGL}" :: RETURN
600 DISPLAY AT(18,11)SIZE(7)
:"CTRL{AADAHAA}" :: RETURN

610 DISPLAY AT(19,11)SIZE(7)
:"CTRL{LLLILLL}" :: RETURN
620 DISPLAY AT(20,11)SIZE(7)
:"CTRL{LLLJLLL}" :: RETURN
630 DISPLAY AT(21,11)SIZE(7)
:"CTRL{AAAKAAA}" :: RETURN
640 DISPLAY AT(22,11)SIZE(7)
:"CTRL{LLGLCLL}" :: RETURN
650 DISPLAY AT(23,11)SIZE(7)
:"CTRL{LGLLLCL}" :: RETURN
660 DISPLAY AT(24,11)SIZE(7)
:"CTRL{FBBBBBE}" :: RETURN
```

SUPERHIRN

Das vorliegende Programm stellt sehr hohe Anforderungen an Ihre Kombinationsgabe und erfordert eine ausgeklügelte Strategie, um die gestellte Aufgabe wenigstens hin und wieder lösen zu können. Der Computer stellt aus neun Farben einen fünfstelligen Farbcode zusammen, in dem eine Farbe auch mehrmals vorkommen kann. Ihre Aufgabe ist es nun, durch geschickte Abfrage diesen Farbcode zu erraten. Die einzelnen Farben werden über die Zahlentasten (1–9) eingegeben. Zur Korrektur einer Farbe kann der Cursor mit "S" und "D" bewegt werden. Die Eingabe einer Farbfolge wird mit der "ENTER"-Taste beendet. Falls eine Farbe richtig erraten wurde und auf das richtige Feld gesetzt wurde, zeigt dies der Computer durch einen blauen Turm an. Falls eine richtige Farbe erraten, aber auf ein falsches Feld plaziert wurde, so wird dies durch einen weißen Turm angezeigt. Sie haben gewonnen, wenn nach der zwölften Abfrage fünf blaue Türme angezeigt werden, andernfalls wird die richtige Farbkombination auf dem Bildschirm angezeigt. Verzagen Sie nicht, wenn Sie die ersten Male nie gewinnen. Das Spiel ist so schwierig, daß auch mit viel Übung meist der Computer gewinnt. Dafür verliert dieses Programm aber auch nie seinen Reiz, denn bei der Fülle der möglichen Farbkombinationen gibt es immer schwierige und neue Möglichkeiten.

ADVENTURE

In diesem Action-Adventure müssen Sie zu einem Vampir vordringen und ihn vernichten, zuvor jedoch durch mehrere Screens die Gefahren unbeschadet durchstehen, die dort lauern (Feuerbälle, Geister, Drachen, Ratten, Fallgruben etc.). Manche Leitern, die man nach unten verlassen hat, kann man nicht wieder besteigen. Aber mehr wird nicht verraten, es ist ja schließlich ein Adventure. Und wie vernichtet man doch gleich einen Vampir???
Die Bedienung des Spiels erfolgt über Joystick(1). Die Richtungen werden über den Knüppel bestimmt, bei gleichzeitigem Druck der Feuertaste wird in die betreffende Richtung gesprungen.

MSX®
REVUE
DAS MAGAZIN FÜR FREUNDE DER KOMPATIBLEN
DM 5,80/ÖS 49/SFR 5,80
IM TEST: Yamaha CX 5
SERVICE: Welcher Drucker für MSX?
IM TEST: Textverarbeitung
SERVICE: Basic-Befehle
IM TEST: Joysticks
SERVICE: Über 20 Seiten MSX-Listings

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *     SUPER-HIRN       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Stefan Richter     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     5810 Bytes       *
28 ! *                      *
29 ! ************************
100 ON WARNING NEXT
110 CALL CLEAR :: CALL SCREEN(2):: FOR I=3 TO 8 :: CALL COLOR(I,16,2):: NEXT I
120 CALL COLOR(2,2,2):: CALL HCHAR(4,6,42,22):: CALL HCHAR(13,6,42,22):: CALL VCHAR(5,6,42,8):: CALL VCHAR(5,27,42,8)
130 DATA 83,85,80,69,82,72,73,82,78
140 RESTORE 130 :: CALL MAGNIFY(2):: J=41 :: FOR I=1 TO 9 :: READ C :: CALL SPRITE(#I,C,I+3,J,41+I*16):: J=J+4 :: NEXT I
150 CALL CHAR(59,"000000007C",64,"3C4299A1A199423C")
160 DISPLAY AT(20,4):"SPIELANLEITUNG? <J><N>" :: DISPLAY AT(24,4):"@ 1985 BY STR;SOFTWARE"
170 CALL TITEL :: CALL CHA
180 CALL FELD :: RANDOMIZE
190 FOR I=1 TO 5 :: CODE(I)=INT(RND*9)+1 :: NEXT I
200 X=177 :: Y=49
210 CALL MAGNIFY(3):: CALL SPRITE(#1,140,16,X,Y)
220 CALL KEY(0,K,S):: IF S=0 THEN 220
230 IF K=68 OR K=83 THEN 260
240 IF K>48 AND K<58 THEN 310
250 IF K=13 THEN 350 ELSE 220
260 IF K=68 THEN 290
270 Y=Y-16 :: IF Y=33 THEN Y=113
280 CALL LOCATE(#1,X,Y):: GOTO 220
290 Y=Y+16 :: IF Y=129 THEN Y=49
300 CALL LOCATE(#1,X,Y):: GOTO 220
310 A=(X-1)/8+1 :: B=(Y-1)/8+1 :: Z=(K-48)*8+56
320 CALL HCHAR(A,B,Z):: CALL HCHAR(A+1,B,Z+1):: CALL HCHAR(A,B+1,Z+2):: CALL HCHAR(A+1,B+1,Z+3)
330 Y=Y+16 :: IF Y=129 THEN Y=49
340 GOTO 210
350 CALL DELSPRITE(#1):: RICH,RIC=0 :: FOR I=1 TO 5 :: CO(I),CR(I)=0 :: NEXT I
360 A=(X-1)/8+1 :: B=7 :: FOR I=1 TO 5 :: CALL GCHAR(A,B,Z):: IF Z=136 THEN 210
370 Z=Z/8-7 :: COD(I)=Z :: B=B+2 :: NEXT I
380 FOR I=1 TO 5 :: IF COD(I)<>CODE(I)THEN 400
390 RICH=RICH+1 :: CO(I)=2 :: CR(I)=2
400 NEXT I
410 FOR I=1 TO 5 :: FOR J=1 TO 5
420 IF CR(I)=2 THEN 470
430 IF CO(J)=2 THEN 460
440 IF COD(J)<>CODE(I)THEN 460
450 RIC=RIC+1 :: CO(J)=2 :: GOTO 470
460 NEXT J
470 NEXT I
480 B=20
490 IF RICH=0 THEN 510
500 FOR I=1 TO RICH :: CALL HCHAR(A,B,76):: CALL HCHAR(A+1,B,77):: CALL HCHAR(A,B+1,78):: CALL HCHAR(A+1,B+1,79):: B=B+2 :: NEXT I
510 FOR I=1 TO RIC :: CALL HCHAR(A,B,116):: CALL HCHAR(A+1,B,117):: CALL HCHAR(A,B+1,118):: CALL HCHAR(A+1,B+1,119):: B=B+2 :: NEXT I
520 IF RICH=5 THEN 590
530 X=X-16 :: IF X=-15 THEN 550
540 GOTO 210
550 A=1 :: B=18 :: FOR I=1 TO 5 :: Z=CODE(I)*8+56
560 CALL HCHAR(A,B,Z):: CALL HCHAR(A+1,B,Z+1):: CALL HCHAR(A,B+1,Z+2):: CALL HCHAR(A+1,B+1,Z+3)
570 A=A+2 :: NEXT I
580 GOTO 640
590 FOR I=0 TO 9 :: CALL SOUND(-250,391,I):: NEXT I
600 FOR I=0 TO 9 :: CALL SOUND(-250,329,I,391,I+11):: NEXT I
```

LISTINGS

```
610 FOR I=0 TO 9 :: CALL SOU
ND(-250,261,I,329,I+11,391,I
+21):: NEXT I
620 FOR I=0 TO 9 :: CALL SOU
ND(-250,391,I,261,I+11,329,I
+21):: NEXT I
630 FOR I=0 TO 9 :: CALL SOU
ND(-250,261,I,391,I+11,329,I
+21):: NEXT I
640 CALL KEY(0,K,S):: IF S=0
 THEN 640 ELSE 180
650 SUB TITEL
660 DATA 2,330,2,330,2,349,2
,392,2,392,2,349,2,330,2,294
,2,262,2,262,2,294,2,330,3,3
30,1,294
670 DATA 3,294,2,330,2,330,2
,349,2,392,2,392,2,349,2,330
,2,294,2,262,2,262,2,294,2,3
30,3,294
680 DATA 1,262,3,262,2,294,2
,294,2,330,2,262,2,294,1,330
,1,349,2,330,2,262,2,294,1,3
30,1,349
690 DATA 2,330,2,294,2,262,2
,294,3,196,2,330,2,330,2,349
,2,392,2,392,2,349,2,330,2,2
94,2,262
700 DATA 2,262,2,294,2,330,3
,294,1,262,3,262,0,0
710 I=3
720 RESTORE 660
730 READ D,T :: IF D=0 THEN
720 :: CALL SOUND(D*160,T,4,
T*2,4,T*3,4)
740 CALL COLOR(2,I,I+2):: I=
I+1 :: IF I=15 THEN I=3
750 CALL KEY(0,K,S):: IF S=0
 THEN 730
760 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL CHARSET
770 IF K=74 OR K=106 THEN CA
LL SPANL
780 SUBEND
790 SUB SPANL
800 CALL CLEAR :: FOR I=1 TO
 8 :: CALL COLOR(I,16,2):: N
EXT I
810 DISPLAY AT(1,10):"SUPERH
IRN" :: DISPLAY AT(2,10):"==
========"
820 DISPLAY AT(4,1):"DU HAST
 12 VERSUCHE DAFUER, DEN COD
E ZU KNACKEN.DU MUSST9 FARBE
N AUF 5 FELDER VER-..TEILEN.
 EINE FARBE KANN AUCH"
830 DISPLAY AT(8,1):"MEHRMAL
S VORHANDEN SEIN. ES GIBT 59
049 VERSCHIEDENE.....MOEGLIC
HKEITEN VON DENEN DERCOMPUTE
R EINE GANZ ZUFAELLIG"
840 DISPLAY AT(12,1):"AUSWAE
HLT. HAST DU ES NACH..12 VER
SUCHEN NICHT GESCHAFFTGIBT D
ER COMPUTER SEINEN....CODE B
EKANNT. MOGELN KANN"
850 DISPLAY AT(16,1):"MAN NI
CHT.DER COMPUTER PASSTAUF."
860 DISPLAY AT(19,1):"NUN ZU
R STEUERUNG:.........<1-9>.
.FARBAUSWAHL.........<S/D>.
.LINKS/RECHTS.........ENTER.
.UEBERGABE"
870 DISPLAY AT(24,22):"<ENTE
R>"
880 CALL KEY(0,K,S):: IF S=0
 THEN 880
890 CALL CLEAR :: DISPLAY AT
(10,1):"DU ERHAELST EINEN BL
AUEN....TURM FUER EIN FARBFE
LD DAS..AN DER RICHTIGEN POS
ITION...IST."
900 DISPLAY AT(15,1):"DU ERH
AELST EINEN WEISSEN...TURM F
UER EIN FARBFELD DAS..IM COD
E ENTHALTEN IST, ABER AN DER
 FALSCHEN STELLE"
910 DISPLAY AT(19,1):"LIEGT.
"
920 DISPLAY AT(22,1):"FUER E
IN NEUES SPIEL DRUECKTMAN IR
GEND EINE TASTE. .....VIEL S
PASS!.........<ENTER>"
930 CALL KEY(0,K,S):: IF S=0
 THEN 930
940 SUBEND
950 SUB FELD
960 CALL CLEAR :: CALL COLOR
(2,11,1,3,16,2,4,16,2,5,15,1
,6,5,1,7,7,1,8,11,1)
970 CALL COLOR(9,10,1,10,13,
1,11,16,1,12,14,1,13,4,1,14,
16,2)
980 FOR I=1 TO 23 STEP 2 ::
FOR J=7 TO 15 STEP 2
990 CALL HCHAR(I,J,136):: CA
LL HCHAR(I+1,J,137):: CALL H
CHAR(I,J+1,138):: CALL HCHAR
(I+1,J+1,139)
1000 NEXT J :: NEXT I
1010 FOR I=1 TO 12 :: DISPLA
Y AT(25-2*I,2)SIZE(-2):USING
 "##":I :: NEXT I
1020 SUBEND
1030 SUB CHA
1040 DATA 007F7F7F7F7F7F7F7F
7F7F7F7F7F7F0000FEFEFEFEDEDE
DEDEDEDEDEFEFEFE00,007F7F7F7
F7C7F7F7C7D7D7C7F7F7F0000FEF
EFEFE1EDEDE1EFEFE1EFEFEFE00
1050 DATA 007F7F7F7F7C7F7F7C
7F7F7C7F7F7F0000FEFEFEFE1EDE
```

```
DE1EDEDE1EFEFEFE00,007F7F7F7
F7D7D7D7C7F7F7F7F7F7F0000FEF
EFEFEDEDEDE1EDEDEDEFEFEFE00
1060 DATA 007F7F7F7F7C7D7D7C
7F7F7C7F7F7F0000FEFEFEFE1EFE
FE1EDEDE1EFEFEFE00,007F7F7F7
F7C7D7D7C7D7D7C7F7F7F0000FEF
EFEFE1EFEFE1EDEDE1EFEFEFE00
1070 DATA 007F7F7F7F7C7F7F7F
7F7F7F7F7F7F0000FEFEFEFE1EDE
DEDEDEDEDEFEFEFE00,007F7F7F7
F7C7D7D7C7D7D7C7F7F7F0000FEF
EFEFE1EDEDE1EDEDE1EFEFEFE00
1080 DATA 007F7F7F7F7C7D7D7C
7F7F7C7F7F7F0000FEFEFEFE1EDE
DE1EDEDE1EFEFEFE00,FF8080808
0808080808080808080FFFF010
1010101010101010101010101FF
1090 CALL CLEAR :: RESTORE 1
040 :: FOR I=64 TO 136 STEP
8 :: READ C$ :: CALL CHAR(I,
C$):: NEXT I
1100 CALL CHAR(140,"0000000F
08080808F8080808080F00008080
80F8080808080F08080808F88080
")
1110 FOR I=76 TO 116 STEP 40
 :: CALL CHAR(I,"00002A3F3F3
F0F0F0F0F0F0F0F0F3F7F0000AAF
EFEFEF8F8F8F8F8F8F8F8FEFF"):
: NEXT I
1120 SUBEND
```

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *   ADVENTURE  III     *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *  Martin Hornschuh    *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    11093 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CLEAR :: CALL SCREE
N(2):: CALL MAGNIFY(3):: CAL
L COLOR(1,7,1,2,16,1):: FOR
I=3 TO 8 :: CALL COLOR(I,16,
1):: NEXT I :: FOR I=9 TO 12
 :: CALL COLOR(I,16,7):: NEX
T I
110 CALL CHAR(128,"071F38736
4E8C8C8C8C8E86473381F07E0F81
C8E4627030303032746 8E1CF8E",
33,RPT$("F",16),40,RPT$("F",
16))
120 PRINT "...((( ((..( ( ((
( (..(":"...( ( ( ( ( (...
(..(":"...((( ( ( ( ( ((..((
 ( (((":"...( (.( ( ( ( (...
( (("
130 PRINT "...( ( ((...(..((
( (..(": :"!!!!.............
.......!!!!....! ((( ( (..((
( ((( !.........!..(..( (..(
( (...!"
140 PRINT "....!..(..( (..((
( ((..!.........!..(..( (..((
..(...!.........!..(..(((..(
( ((( !.........!............
.......!"
150 PRINT ".....!!!!der!vamp
ir!!!!": : :".......1985 BY"
:".......BONISOFTWARE": :"..
.SPIELANLEITUNG ?..J - N"
160 CALL SPRITE(#1,128,3,154
,54)
170 FOR I=16 TO 3 STEP -1 ::
 CALL COLOR(#1,I):: CALL KEY
(0,K,S):: IF K=74 THEN 190 E
LSE IF K=78 THEN 300
180 CALL SOUND(-1000,RND*110
0+110,10,RND*1200+110,12)::
NEXT I :: GOTO 170
190 CALL DELSPRITE(#1):: CAL
L CLEAR :: PRINT "IN DEM SPI
EL >> ADVENTURE <<GEHT ES DA
RUM IM LAND DER...ELFEN UND
```

BÖRSE

TI 99/4 A: Verk. folgende Module: Buck Rogers 40,-- DM, Congo Bongo 50,-- DM (auch Tausch): Alligator Mix , 10,-- DM: Jungle Hunt 50,-- DM: Bei M. Schenk, Rosenweg 4, 7107, Bad Friedrichshall.

Suche in Assembler (!) geschriebenes Editor/Assembler/Disassembler-Paket für Mini-Memory + 32 K sowie für Ex Basic + 32 K. Suche auch Assembler-Prgr. auf Kassette. Alfred Manthey-Rojas, Eichenkamp 22, 2200 Elmshorn, 04121-74734.

TI 99/4A m. Ext-Basic, Recorder + Lit. 380-Module: Pole Position + Jungle Hunt a 50,-- DM, Shamus + Protector a 35,-- DM, TI Invaders, Munchman, Hunt the Wumpus, Jaw breaker, The Attack, Tombstone City, Chisholm Trail a 25,--, Car tridge Expander 3-fach, 80,-- DM. H. Menzel, I.d. Freßäckern 17/1, 7120 Bietigheim-Bissg.

Kaufe TI 99-Konsole, Angebote an: Heinz-J. Fischer, Bergisch-Gladbacherstr. 800, 5000 Köln 80.

TI 99/4A – 100 DM. Mini Memory – 150 DM, TI Forth 75 DM, Alle Preise VB. Tel.: 0814194897.

TI Austausch. A. Münzenmaier. Habe deine Adresse verlegt. Bitte melden bei K. Wolters: 0201/772141.

Verkaufe: Module Disk Managr. 2, und Schach je DM 50,--, 3-fach-Modulexpander Dm 75,--. Lucas, 06821/70160.

Top-Angebot: Datenverw. & Analyse, Text & Datei, Invaders, D-Kong, Pac-Man, alles Original Module, günstig, Tel.: 089/ 3513935.

Adventure-Modul mit allen Adventures (100) sowie div. andere Module günstig abzugeben. Tel.: 089/ 3513935.

```
FEEN EINEN VAMPIR ŻU KILLEN,
 DER ALS ERSTER"
200 PRINT "SEINER RASSE AUCH
 IM BLANKENSONNENLICHT UEBER
LEBEN KANN. ER KOENNTE DESHA
LB DEM GE- SAMTEN LAND GEFAE
HRLICH WER"
210 PRINT "DEN. ALSO WURDEST
 DU VON....DEINEN >> FREUNDE
N << DAZU..VERDONNERT DEN VA
MPIR ZU....TOETEN. DU GEHST
ALSO LOS"
220 PRINT "UND MERKST, DASS
 DER VAMPIR UEBERALL WAECHTER
 UND AEHN- LICHES AUFGESTELL
T HAT, DIE DIR SELTSAMERWEIS
E GEFAEHR-"
230 PRINT "LICH WERDEN KOENN
TEN.": : :" >> BITTE TASTE D
RUECKEN <<"
240 CALL KEY(0,K,S):: IF S=0
 THEN 240
250 CALL CLEAR :: PRINT "ES
HEISST NUN FUER DICH DEN GEF
AHREN AUSZUWEICHEN UND ZUDEM
 VAMPIR ZU GELANGEN. ABERWIE
 TOETETE MAN DOCH GLEICH"
260 PRINT "EINEN VAMPIR ????
??????????": : :"STEUERUNG:
": : :"GEHEN...: JOYSTICK RE
CHTS.............ODER LINKS"
: :
270 PRINT "SPRINGEN: JOYSTIC
K RECHTS.............ODER LI
NKS UND..............DAZU AC
T. - KNOPF...........DRUECKE
N."
280 PRINT : : :"> TASTE ZUM
START DRUECKEN <"
290 CALL KEY(0,K,S):: IF S=0
 THEN 290
300 CALL CLEAR :: PRINT "WUR
DE DIE >> ALPHA LOCK << -TAS
TE AUSGERASTET ? N"
310 CALL DELSPRITE(#1):: ACC
EPT AT(23,21)BEEP VALIDATE("
j")SIZE(-1):E$ :: IF E$<>"j"
 THEN 310
320 CALL CLEAR :: RANDOMIZE
:: CALL DELSPRITE(#1):: CALL
 SCREEN(5):: CALL COLOR(6,2,
1):: BI=0 :: L=3 :: ON ERROR
 350
330 RESTORE :: FOR I=40 TO 1
27 :: READ A$ :: CALL CHAR(I
,A$):: NEXT I
340 BI=BI+1
350 ON BI GOTO 360,450,580,5
70,740
360 Z=INT(RND*100):: IF Z<50
 OR Z>100 THEN 360
```

LISTINGS

```
370 CALL COLOR(12,9,9,13,7,7
):: CALL HCHAR(13,1,120,7*32
):: CALL HCHAR(20,1,128,5*32
)
380 CALL SPRITE(#27,52,9,81,
RND*222+1,0,-1,#28,52,9,81,R
ND*50+10,0,-1,#1,40,2,137,12
0,#2,48,13,137,40)
390 FOR I=1 TO 10 :: CALL CO
INC(ALL,T):: IF T THEN 380
400 NEXT I
410 CALL SPRITE(#3,48,8,137,
98,#4,48,12,137,180,#25,56,1
5,30,RND*200+1,0,3,#26,56,16
,50,RND*20+1,0,4-RND)
420 FOR I=2 TO 4 :: CALL MOT
ION(#I,0,-6):: NEXT I :: X1=
120 :: Y1=137
430 CALL PATTERN(#1,40):: CA
LL KEY(2,K,S):: IF K=18 THEN
 CALL SP1(X1,Y1)ELSE CALL CO
INC(ALL,T)
440 CALL PATTERN(#1,44):: IF
 T THEN GOSUB 1050 ELSE CALL
 SOUND(-100,-7,5):: Z1=Z1+1
:: IF Z1>Z THEN 340 ELSE 430
450 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL COLOR(5,7,1
,12,7,7):: FOR I=1 TO 16 ::
CALL HCHAR(I,17,120,16):: NE
XT I
460 CALL HCHAR(23,1,120,64):
: DISPLAY AT(23,1)SIZE(28):"
xxxACxxACxACxxACxACxxxxxACxx
"
470 FOR I=2 TO 7 :: CALL HCH
AR(I,23,32,10):: NEXT I :: F
OR I=8 TO 16 :: CALL HCHAR(I
,24,32,7):: NEXT I
480 FOR I=7 TO 22 :: DISPLAY
 AT(I,22)SIZE(7):"HIJxHIJ" :
: NEXT I :: CALL HCHAR(7,27,
32):: CALL VCHAR(17,27,32,6)
:: CALL HCHAR(23,31,32,2)
490 CALL SPRITE(#1,40,2,161,
6,#2,48,7,50,217,#3,48,7,190
,185):: X1=6 :: Y1=161
500 CALL MOTION(#2,0,0,#3,0,
0):: CALL SPRITE(#4,68,13,16
1,25,0,3)
510 CALL JOYST(2,X,Y):: CALL
 KEY(2,K,S):: IF K=18 THEN C
ALL SP2(X1,Y1,X,T)ELSE X1=X1
+2*X :: CALL LOCATE(#1,Y1,X1
):: IF X<>0 THEN CALL SOUND(
-10,-7,0)
520 CALL GCHAR(Y1/8+3,X1/8+1
,A):: CALL COINC(ALL,T):: IF
 T OR A<>120 THEN GOSUB
1050 ELSE IF X1>189 AND Y=4
THEN 530 ELSE 510
```

LISTINGS

```
530 CALL GCHAR(Y1/8+1,X1/8+1
,Q):: IF Q<>73 THEN 510 ELSE
 CALL DELSPRITE(#4):: CALL M
OTION(#2,20,0,#3,20,0)
540 CALL JOYST(2,X,Y):: Y1=Y
1+2*-Y :: CALL LOCATE(#1,Y1,
X1):: IF (X<>0)*(Y1=161)THEN
 500 ELSE CALL KEY(2,K,S)::
IF K=18 THEN CALL SP2(X1,Y1,
X,T)
550 CALL COINC(ALL,T):: IF T
 THEN GOSUB 1050 ELSE IF Y<>
0 THEN CALL SOUND(-10,-3,5)
560 CALL PATTERN(#1,40):: IF
 Y1=41 THEN 340 ELSE CALL GC
HAR(Y1/8+1,X1/8+1,Q):: IF Q=
120 OR Q=32 THEN GOSUB 1050
ELSE 540
570 GOSUB 900 :: GOTO 630
580 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL COLOR(5,16,
1,12,16,16):: FOR I=1 TO 5 :
: PRINT TAB(25);"HIJ" :: NEX
T I
590 PRINT " HIJxxACxACxxACxA
CxxACxAxxx..HIJxxxxxxxxxxxxx
xxxxxxxxxx"
600 FOR I=1 TO 6 :: PRINT "
HIJ" :: NEXT I :: PRINT " xx
xxxACxxACxACxxACxACxxHIJ..xx
xxxxxxxxxxxxxxxxxxxxxxHIJ"
610 FOR I=1 TO 6 :: PRINT TA
B(25);"HIJ" :: NEXT I :: PRI
NT "xxxxxxxACxACxxACxACxxACx
xxx xxxxxxxxxxxxxxxxxxxxxxxx
xxx"
620 CALL SPRITE(#1,76,2,153,
20,#2,80,11,75,200,0,-20,#3,
84,7,162,180,0,5,#4,88,15,34
,10,0,-20):: X1=20 :: Y1=153
630 CALL JOYST(2,X,Y):: CALL
 KEY(2,K,S):: IF K=18 THEN C
ALL SP2(X1,Y1,X,T)ELSE X1=X1
+2*X :: CALL LOCATE(#1,Y1,X1
):: CALL GCHAR(Y1/8+3,X1/8+1
,A)
640 CALL GCHAR(Y1/8+2,X1/8+1
,B):: CALL COINC(ALL,T):: IF
 B=73 THEN 660 ELSE IF T OR(
A<>120)*(A<>72)*(A<>73)*(A<>
74)THEN GOSUB 1050 ELSE IF A
=73 THEN 660
650 IF X<>0 THEN CALL SOUND(
-50,-7,5):: GOTO 630 ELSE 63
0
660 IF BI=4 THEN 700 ELSE IF
 Y=0 THEN 630
670 CALL JOYST(2,X,Y):: Y1=Y
1+2*-Y :: IF Y1<10 THEN 340
ELSE CALL LOCATE(#1,Y1,X1)::
 CALL GCHAR(Y1/8+2,X1/8+1,A)
:: IF A<>73 THEN 630
680 CALL COINC(ALL,T):: IF T
 THEN GOSUB 1050 ELSE IF Y<>
0 THEN CALL SOUND(-50,-5,5)
690 GOTO 670
700 IF Y=0 THEN 630
710 CALL JOYST(2,X,Y):: Y1=Y
1+2*-Y :: IF Y1<5 THEN 340 E
LSE CALL LOCATE(#1,Y1,X1)::
CALL GCHAR(Y1/8+3,X1/8+1,A):
: IF A=120 THEN 630
720 IF A=32 THEN Y1=Y1-16 EL
SE IF Y<>0 THEN CALL SOUND(-
50,-5,5)ELSE CALL COINC(ALL,
T):: IF T THEN GOSUB 1050
730 GOTO 710
740 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL COLOR(5,11,
1,6,11,1,10,7,1,11,7,7,12,11
,11)
750 PRINT ".....pp pp pp
pp.................pp pp pp
pp.................ppppHIJpp
pp.................ppppHIJpp
pp"
760 PRINT ".....hppHIJpp
j...................ippHIJpp
k...................ipppHIJpp
pk..................hpppHIJpp
pj"
770 PRINT ".....hppppHIJpp
ppj.................ippppHIJpp
ppk.................ipppppHIJpp
pppk................hpppppHIJpp
pppj"
780 PRINT "HIJxACxACxxACxACx
xACxACxHIJ HIJxxxxxxxxxxxxxx
xxxxxxHIJ" :: FOR I=1 TO 6
:: PRINT "HIJ";TAB(25);"HIJ"
 :: NEXT I
790 PRINT "xxxxxACxxACxHIJxA
CxxACxACxxxxxxxxxxxxxxxHIJxx
xxxxxxxxxxx" :: CALL SPRITE(
#1,76,2,169,117,#2,100,8,8,1
17):: X1=117 :: Y1=169
800 CALL SPRITE(#3,84,7,162,
50,0,10,#4,80,15,140,200,0,-
10)
810 FOR I=1 TO 4 :: CALL SPR
ITE(#I+4,48,I+10,200,I+20*I,
#I+8,48,I+4,230,130+I+20*I):
: NEXT I
820 CALL JOYST(2,X,Y):: CALL
 KEY(2,K,S):: IF K=18 THEN C
ALL SP2(X1,Y1,X,T)ELSE X1=X1
+2*X :: CALL LOCATE(#1,Y1,X1
):: CALL GCHAR(Y1/8+3,X1/8+1
,A)
830 CALL GCHAR(Y1/8+2,X1/8+1
,B):: IF B=73 THEN 850 ELSE
IF T+(A=32)+(A=65)+(A=67)THE
```

```
N GOSUB 1050 ELSE IF A=73 TH
EN 850
840 CALL COINC(ALL,T):: IF X
<>0 THEN CALL SOUND(-50,-7,5
):: GOTO 820 ELSE 820
850 IF Y=0 THEN 820
860 CALL JOYST(2,X,Y):: Y1=Y
1+2*-Y :: IF Y1<20 THEN GOSU
B 1050 ELSE CALL LOCATE(#1,Y
1,X1)
870 IF Y1=89 THEN 890 ELSE C
ALL GCHAR(Y1/8+2,X1/8+1,A)::
 IF A=32 THEN 820 ELSE IF Y<
>0 THEN CALL SOUND(-50,-6,5)
880 GOTO 860
890 CALL DELSPRITE(#3,#4)::
FOR I=4 TO 12 :: CALL MOTION
(#I,RND*5+2,0):: NEXT I :: G
OTO 820
900 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL COLOR(5,15,
1,12,15,15):: PRINT "xxxxxxx
xxxxxxxxHIJxxxxxxxxxxxxxxxxx
xxxxxxxxHIJxxxxxxxxxx"
910 FOR I=1 TO 19 :: PRINT T
AB(16);"HIJ" :: NEXT I :: PR
INT "xxxACxACxxACxACxxxxACxx
x....xxxxxxxxxxxxxxxxxxxxxx
x"
920 FOR I=1 TO 4 :: CALL SPR
ITE(#I+1,80,I+10,I+32*I,RND*
200+1,0,-10,#I+5,92,6+I,I+32
*I+10,RND*200+2,0,10)
930 NEXT I :: CALL SPRITE(#1
,76,2,153,20,#28,96,11,153,1
90):: X1=20 :: Y1=153 :: RET
URN
940 DATA 0103030103OF1F37,27
07060E1C181,80C0C084CCF8F0E,
E0E060606060607,01030321331F
0F07,0707060606060607,80C0C0
80C0F0F8EC,E4E060341C0C
950 DATA ,,,0844B41B7C942412
,,0000000030F1F7F,,00000030
78FEFEFF,000000030F1F7F0F,03
,00003078FEFEFFFE,FCE
960 DATA 0103030103OF3F27,07
07063E3C2,80C0C080C4FCF8E0,E
0E0647C3C
970 DATA ,80C0E0F0F8FCFEFF,,
0103070F1F3F7FFF,0000A040234
F9EB8,FC7E777370484476,00000
00088FFFC1E,0020FCF078
980 DATA 0606060707060606,00
0000FFFF,606060E0E060606,,01
030301030F1F37,2707060606060
60E,80C0C080C0F0F8EC,E4E0606
06060607
990 DATA 3C7EFFB76726060C,,0
000B1FA44,,0000070F9C6101,,1
038ECFFF8B0B,,081C37FF1F0D0D
,,0000E0F0C9868,
1000 DATA 0000032F11000103,,
000C3FF8EEE0C08,,0103030103 3
57F35,0301010101010307,00808
0008058FC58,8000000000080C
1010 DATA 01030B0D1F3F7F47,2
717160E0E06060E,80C0D0B0F8FC
FEE2,E4E868707060607,7F7F7F7
F7F7F7F7F,FFFFFFFFFFFFFFFF,F
EFEFEFEFEFEFEFE,FFFFFFFFFFFF
FFFF
1020 DATA 0000000900200008,00
40001004,0000000020080002,00
0800208,0000000014004,800020
0108,0000000000802004,000410
4
1030 DATA ,0200000001040203,
0000000000004,0000088000A0C0
C,000000000004,01080C0204120
F01,0008000020000008,089450D0
94E8F0C
1040 DATA 0210000400100A09,0
602030602090601,00900240C4C8
6034,32326444E7ECF0F
1050 CALL PATTERN(#1,76):: I
F BI=4 AND X1>179 THEN CALL
DELSPRITE(#28):: KR=1 :: RET
URN
1060 IF BI=1 THEN 1070 ELSE
1080
1070 FOR I=2 TO 4 :: CALL DE
LSPRITE(#I):: NEXT I :: CALL
 MOTION(#27,0,0,#28,0,0)
1080 IF (BI=5)*(KR=1)*(Y1<20
)THEN CALL S
1090 CALL SPRITE(#20,112,11,
Y1-6,X1)
1100 FOR I=1 TO 10 :: FOR J=
0 TO 4 STEP 4 :: CALL PATTER
N(#20,108+J):: CALL SOUND(-1
00,110+I*RND+J,2,-7,5):: CAL
L COLOR(#20,I+J)
1110 NEXT J :: NEXT I :: L=L
-1 :: IF L<>0 THEN CALL DELS
PRITE(ALL):: RETURN 350
1120 CALL SPRITE(#20,116,13,
Y1,X1):: FOR I=1 TO 10 :: FO
R J=0 TO 8 STEP 4 :: CALL PA
TTERN(#20,116+J):: CALL SOUN
D(-100,-7,5,1200-I*J*10,2)
1130 CALL COLOR(#20,RND*10+3
):: NEXT J :: NEXT I
1140 CALL DELSPRITE(ALL):: C
ALL CLEAR :: CALL CHARSET ::
 DISPLAY AT(1,1):"ACH - HAT
DICH DER VAMPIR...GEROESTET
??? DAS TUT MIR"
1150 DISPLAY AT(3,1):"ABER L
EID !!! SOLLTEST EBEN NICHT
SO UNGESCHICKT SEIN.": : :"W
ILLST DU ES NOCH EINMAL....V
```

```
ERSUCHEN? J"
1160 ACCEPT AT(8,12)BEEP VAL
IDATE("JNjn")SIZE(-1):E$ ::
IF E$="J" OR E$="j" THEN 118
0 ELSE CALL CLEAR
1170 PRINT "..ICH HASSE FEIG
LINGE !!!!...----------------
---------- ----------------
----------" :: CALL SOUND(
4250,-7,0):: END
1180 CALL CLEAR :: PRINT " O
H, WELCH MUTIGE UND KLUGE...
.....PERSON !!!!" :: CALL SO
UND(4250,-3,0):: RUN 320
1190 SUB SP1(X1,Y1):: CALL P
ATTERN(#1,60):: CALL COINC(A
LL,T):: IF T THEN SUBEXIT EL
SE CALL LOCATE(#1,Y1-10,X1)
1200 CALL SOUND(-100,-3,0)::
 FOR I=1 TO 14 :: CALL SOUND
(-100,110+I*10,0):: NEXT I :
: CALL LOCATE(#1,137,120)::
CALL PATTERN(#1,40)
1210 CALL COINC(ALL,T):: IF
T THEN SUBEXIT
1220 SUBEND
1230 SUB SP2(X1,Y1,X,T):: IF
 X=0 THEN SUBEXIT ELSE CALL
PATTERN(#1,60):: CALL SOUND(
-100,-3,0)
1240 FOR J=-4 TO 4 STEP 8 ::
 FOR I=1 TO 4 :: X1=X1+X ::
Y1=Y1+J :: CALL LOCATE(#1,Y1
,X1)
1250 CALL COINC(ALL,T):: IF
T THEN SUBEXIT
1260 NEXT I :: NEXT J :: CAL
L PATTERN(#1,76):: X=0 :: SU
BEND
1270 SUB S :: FOR I=1 TO 15
:: CALL SCREEN(15):: CALL SO
UND(50,110+I*10,I,-5,I):: CA
LL SCREEN(16):: CALL SOUND(5
0,110+I*15,I,-6,I)
1280 CALL SCREEN(11):: CALL
SOUND(50,110+I*20,I,-7,I)::
CALL SCREEN(12):: CALL SOUND
(50,110+I*25,I,-3,I):: NEXT
I
1290 CALL DELSPRITE(ALL):: C
ALL CLEAR :: CALL CHARSET ::
 DISPLAY AT(1,1):"HALLO DU M
OERDER !!! HAST...DEN ARMEN
VAMPIR ABER GANZ"
1300 DISPLAY AT(3,1):"SCHOEN
 GEAERGERT - ER HAT...SICH V
OR WUT SELBER ERLE-...DIGT.
ABER MACH DIR NICHTS..DRAUS.
 AM ENDE SOLTE EIGENT-"
1310 DISPLAY AT(7,1):"LICH I
MMER DER GUTE GEWINNEN.....-
 ALSO DU !!!"
1320 DISPLAY AT(10,1):"WILLS
T DU NOCH MEHR VAMPIRE KILLE
N (J / N)?"
1330 ACCEPT AT(11,18)VALIDAT
E("JNjn")BEEP SIZE(1):A$ ::
IF A$="" THEN 1330 ELSE IF A
$="J" OR A$="j" THEN 1350
1340 CALL CLEAR :: PRINT ".
...FEIGLING !!!!.....
...----------" :: CALL SO
UND(4250,-7,0):: END
1350 CALL CLEAR :: PRINT " O
H, EIN MUTIGER MENSCH !!!! -
----------"
:: CALL SOUND(4250,-1,0):: R
UN 320
1360 SUBEND
```

LISTINGS

IRVEL

Befreien Sie Ihren Freund aus den Fängen von "Racon". Zu Beginn des Spiels werden die verschiedenen Screens solange in Folge gezeigt, bis eine Taste gedrückt wird. Danach befindet man sich in Ebene drei. Von hier aus hat man sich bis in Ebene eins hochzuarbeiten. In jeder Ebene kann man die Bildschirme beliebig oft durchlaufen. Ebenen, die man nach oben verlassen hat, kann man nach unten jedoch nicht mehr betreten. Man kann sich auch innerhalb einer Ebene nach oben (=hinten) und unten (=vorn) bewegen, jedoch ist immer darauf zu achten, daß die Füße auf der blauen Fläche bleiben, sonst gibt's Strafe. In den Ebenen zwei und drei müssen Sie Gegenstände aufnehmen, um zum Kampf in Screen 4 gerüstet zu sein. Es ist an Ihnen herauszufinden, wieviele und welche Gegenstände Sie benötigen, um in die vierte Ebene zu gelangen. Als Hilfe: Gegenstände, die Sie nicht (mehr) brauchen, werden nicht mehr oben im Bildschirm dargestellt, wenn Sie sie aufnehmen, indem Sie mit der Spielfigur darüber hinwegziehen. Durch häufigen Screenwechsel innerhalb einer der beiden oberen Ebenen werden immer neue und andere Gegenstände zur Aufnahme verfügbar! Aber genug verraten.

Haben Sie Ihre Ausrüstung beieinander, so müssen Sie den Weg zu Racon finden. Aber da steht ja ein Tempel! Wählen Sie auf Aufforderung Ihre Waffen! Um Racon vollständig zu besiegen und Ihren Freund zu befreien, genügt ein Durchgang nicht, sondern Sie müssen alle drei Levels bestehen!!!

Achten Sie auf die Vasallen von Racon.

Sie müssen also folgende Aufgaben lösen:

Wie ist die Kombination der Gegenstände?

Wie gelangt man zum Tempel von Racon?

Wie tötet man Racon?

In Zeile 1280 kann folgende Änderung vorgenommen werden:

XY=INT(RND)+8 macht das Spiel sehr schwer, da Gegenstände nun in allen Ebenen erscheinen können.

XY=INT(RND*4)+8 macht das Spiel sehr, sehr einfach, da Gegenstände nur in Ebene 1 (oberste) erscheinen. Die im Programm abgedruckte Zeile ist ein mittlerer Schwierigkeitsgrad.

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *       IRVEL II       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *  Manfred Lipowski    *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! *    Cassettenrec.     *
24 ! *(oder Disk+32k-Erw.)  *
25 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    12368 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CLEAR :: RANDOMIZE
:: CALL MAGNIFY(3):: CALL SC
REEN(2):: CALL CHARPAT(73,A$
,82,A1$,86,A2$,69,A3$,76,A4$
):: CALL CHAR(58,A$,59,A1$,6
0,A2$,61,A3$,62,A4$):: CALL
CHA1
110 CALL HCHAR(1,1,72,64)::
CALL HCHAR(3,1,74):: CALL HC
HAR(3,2,73):: CALL HCHAR(3,3
1,73):: CALL HCHAR(3,32,74):
: CALL HCHAR(4,2,78)
120 CALL HCHAR(5,2,79):: CAL
L HCHAR(6,2,77):: CALL HCHAR
(7,2,78):: CALL HCHAR(4,31,7
9):: CALL HCHAR(4,32,78):: C
ALL HCHAR(5,31,77)
130 CALL HCHAR(6,32,79):: CA
LL HCHAR(7,31,79):: CALL HCH
AR(7,32,78):: CALL HCHAR(1,4
,48,5):: DISPLAY AT(1,22)BEE
P:";;<=>H?" :: A=0
140 A=A+1 :: IF A=8 THEN A=1
 ELSE ON A GOSUB 170,180,190
,200,190,180,170
150 FOR B=1 TO 99 :: CALL KE
Y(1,K,S):: IF K=18 THEN 210
160 NEXT B :: GOTO 140
170 CALL FARB(2):: CALL WALD
 :: CALL MITT :: CALL LINK :
: CALL FARB(1):: RETURN
180 CALL FARB(2):: CALL WALD
 :: CALL MITT :: CALL FARB(1
):: RETURN
190 CALL FARB(2):: CALL TEMP
(0):: CALL MITT :: CALL RECH
 :: CALL FARB(1):: RETURN
200 CALL FARB(2):: CALL TEMP
(1):: CALL LINK :: CALL RECH
 :: CALL FARB(1):: RETURN
210 CALL FARB(2):: CALL HCHA
R(1,4,72,5):: CALL WALD :: C
ALL MITT :: CALL LINK :: CAL
L FARB(1):: CALL SPRITE(#1,9
6,16,160,64):: BILD=1
220 SCHL,SC,SCHW,AXAX=0 :: R
UND=1 :: CALL HCHAR(1,1,72,2
0)
230 CALL PATTERN(#2,132,#3,1
40,#4,132):: CALL SOUND(-1,-
6,0)
240 CALL PATTERN(#1,96):: CA
LL COINC(ALL,ST):: IF ST THE
N 1230
250 CALL JOYST(1,X,Y):: CALL
 POSITION(#1,X1,Y1):: CALL C
OINC(ALL,ST):: IF ST THEN 12
30
260 CALL PATTERN(#2,128,#3,1
36,#4,128):: IF BILD=4 AND X
1=72 THEN 730
270 IF X=-4 THEN 280 ELSE IF
 Y=-4 THEN 290 ELSE IF X=4 T
HEN 300 ELSE IF Y=4 THEN 310
 ELSE IF BILD=2 OR BILD=4 TH
EN 230 ELSE 240
280 Y1=Y1-8 :: PAT=116 :: TO
N=-5 :: GOTO 320
290 X1=X1+8 :: PAT=100 :: TO
N=-6 :: GOTO 320
300 Y1=Y1+8 :: PAT=108 :: TO
N=-5.23 :: GOTO 320
310 X1=X1-8 :: PAT=100 :: TO
N=-7
320 IF Y1<16 OR Y1>240 THEN
580
330 CALL PATTERN(#1,PAT):: C
ALL GCHAR(X1/8+2,Y1/8+1,AS):
: IF AS<>80 THEN 340 ELSE 40
0
340 IF AS=36 OR AS=37 THEN 4
00 ELSE IF AS=34 THEN 420 EL
SE IF AS>=124 AND AS<=127 TH
EN 460
350 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,X1,Y1):: CALL P
ATTERN(#1,88):: CALL SOUND(-
9,-5,0):: CALL COLOR(#1,15):
: B=0
360 A=4 :: B=B+2 :: CALL SOU
ND(-300,800-B*3,9,700-B*3,9)
:: CALL POSITION(#1,X1,Y1)::
 CALL COLOR(#1,16):: CALL LO
CATE(#1,X1+A,Y1)
370 IF X1>=168 THEN 380 ELSE
 CALL COLOR(#1,15):: GOTO 36
0
380 FOR A=0 TO 25 STEP 5 ::
CALL SOUND(-99,-5,A,110,A,11
2,A):: CALL COLOR(#1,16):: C
ALL SOUND(-99,-7,A+2,110,A,1
12,A,220,A):: CALL COLOR(#1,
2)
```

```
390 NEXT A :: CALL ZE(999)::
 GOTO 210
400 CALL PATTERN(#2,132,#3,1
40,#4,132)
410 CALL SOUND(1,TON,6):: CA
LL LOCATE(#1,X1,Y1):: CALL P
ATTERN(#1,PAT+4):: CALL SOUN
D(1,TON,9):: GOTO 250
420 IF SCHL<1 OR AXAX<=3 OR
SCHW<=1 THEN 350 ELSE CALL H
CHAR(1,11,72):: CALL SOUND(2
5,-5,0)
430 CALL LOCATE(#1,X1,Y1)::
CALL PATTERN(#1,96):: CALL S
OU(1):: CALL SOU(2):: CALL Z
E(555):: CALL SOUND(4250,-4,
0):: CALL FARB(2)
440 CALL MITT :: CALL TEMP(1
):: BILD=4 :: CALL LINK :: C
ALL RECH :: CALL LOCATE(#1,1
60,48):: CALL FARB(1):: SC=S
C+1000 :: CALL SCO(SC)
450 GOSUB 670 :: GOTO 250
460 CALL LOCATE(#1,X1,Y1)::
IF AS=124 THEN 470 ELSE 490
470 SCHL=SCHL+1 :: IF SCHL=1
 THEN CALL HCHAR(1,11,124)::
 SC=SC+250 :: CALL SCO(SC)::
 GOTO 570
480 IF SCHL>1 THEN CALL HCHA
R(1,11,72):: SC=SC-250 :: CA
LL SCO(SC):: SCHL=0 :: GOTO
570
490 IF AS=126 THEN 500 ELSE
510
500 SC=SC+50 :: CALL SCO(SC)
:: GOTO 570
510 IF AS=125 THEN 520 ELSE
540
520 SCHW=SCHW+1 :: IF SCHW>2
 THEN 530 ELSE CALL HCHAR(1,
11+SCHW,125):: SC=SC+100 ::
CALL SCO(SC):: GOTO 570
530 SC=SC+50 :: CALL SCO(SC)
:: GOTO 570
540 IF AS=127 THEN 550 ELSE
570
550 AXAX=AXAX+1 :: IF AXAX>4
 THEN 560 ELSE CALL HCHAR(1,
13+AXAX,127):: SC=SC+125 ::
CALL SCO(SC):: GOTO 570
560 SC=SC+50 :: CALL SCO(SC)
570 CALL HCHAR(X1/8+2,Y1/8+1
,80):: CALL SOUND(5,-5,0)::
GOTO 250
580 IF Y1>=240 AND BILD=1 TH
EN 590 ELSE 610
590 CALL FARB(2):: CALL WALD
 :: CALL MITT :: BILD=2 :: C
ALL LOCATE(#1,X1,16):: CALL
FARB(1):: CALL SCHA :: CALL
MONS(X1,1,RUND)
600 GOTO 250
610 IF Y1<=16 AND BILD=2 THE
N 620 ELSE IF Y1>=240 AND BI
LD=2 THEN 630 ELSE 640
620 CALL FARB(2):: CALL MITT
 :: CALL LINK :: BILD=1 :: C
ALL LOCATE(#1,X1,240):: CALL
 FARB(1):: CALL SCHA :: GOTO
 250
630 CALL FARB(2):: CALL MITT
 :: CALL RECH :: CALL TEMP(0
):: BILD=3 :: CALL LOCATE(#1
,X1,16):: CALL FARB(1):: CAL
L SCHA :: GOTO 250
640 IF Y1<=16 AND BILD=3 THE
N 650 ELSE 660
650 CALL FARB(2):: CALL WALD
 :: CALL MITT :: BILD=2 :: C
ALL LOCATE(#1,X1,240):: CALL
 FARB(1):: CALL SCHA :: CALL
 MONS(X1,2,RUND)
660 GOTO 250
670 CALL LOCATE(#27,89,184,#
28,137,64)
680 CALL CHAR(128,"013B7FFBF
BBBBABBBE5F6F777B3D1E1FC0A0E
0F08020E0C000008CD0F8FC7A0A"
)
690 CALL CHAR(132,"013B7FFFF
BBBBABBBE5F6F77783F1F1FC0A0E
0F0800030E038FCFEE5C5800000"
)
700 CALL CHAR(136,"0F1F1F1F1
7190E0F0F1E3D7B7B7AF7D780C0E
0F0F8FCDE2E7CF8F0E0C040F078"
)
710 CALL CHAR(140,"1F1F1F0F0
F060303070F1F3E3D387F7780C06
0B0D8EC6EEF0FBE7CF8F8F03EAF"
)
720 CALL SPRITE(#2,128,4,57,
62,0,0,#3,136,4,73,62,0,0)::
 CALL HCHAR(11,6,124):: CALL
 SOU(1):: RETURN
730 CALL PATTERN(#1,120):: C
ALL SOUND(5,-5,5)
740 FOR A=110 TO 150 STEP 2
:: B=INT(RND*8)+4 :: CALL SO
UND(-209,A,B+1,A+10,B,300-A,
B-1):: NEXT A
750 CALL CHAR(88,"FF81818181
8181FF",89,"",90,"",91,""):
 CALL SPRITE(#4,88,16,1,96,0
,0):: CALL SOUND(9,-3,0)
760 CALL JOYST(1,X,Y):: CALL
 POSITION(#4,X2,Y2):: CALL S
OUND(1,-5,26)
770 CALL KEY(1,K,S):: IF K=1
8 THEN 800
780 IF X=-4 THEN Y2=Y2-8 ELS
```

LISTINGS

```
E IF X=4 THEN Y2=Y2+8 ELSE I
F X=0 AND Y=0 THEN 760
790 CALL LOCATE(#4,X2,Y2)::
GOTO 760
800 CALL SOUND(9,220,0,222,0
):: CALL GCHAR(X2/8+1,Y2/8+1
,LP):: CALL COLOR(#4,1):: IF
 LP=125 THEN 810 ELSE IF LP=
127 THEN 820 ELSE 1080
810 CALL CHAR(89,"02FF02")::
 YU=1 :: GOTO 830
820 CALL CHAR(89,"7F30FC783"
)):: YU=2
830 CALL CHAR(88,"",90,"",91
,"")
840 CALL POSITION(#1,X1,Y1):
: CALL LOCATE(#4,X1,Y1-7)::
CALL COLOR(#4,16):: B=2+RUND
*2
850 CALL JOYST(1,X,Y):: CALL
 POSITION(#1,X1,Y1,#4,X2,Y2)
860 CALL KEY(1,K,S):: IF K=1
8 THEN 1000
870 IF X=-4 THEN 880 ELSE IF
 X=4 THEN 890 ELSE IF X=0 AN
D Y=0 THEN 900
880 Y1=Y1-8 :: PAT=116 :: Y2
=Y2-8 :: GOTO 930
890 Y1=Y1+8 :: PAT=108 :: GO
TO 930
900 CALL PATTERN(#2,128,#3,1
36):: CALL SOUND(5,-5,0):: C
ALL POSITION(#2,A1,B1,#3,A2,
B2)
910 CALL LOCATE(#2,A1,B1+B,#
3,A2,B2+B):: CALL PATTERN(#2
,132,#3,140):: CALL SOUND(5,
-6,1):: CALL COINC(#3,#1,15,
PL):: IF PL THEN 1080
920 GOTO 850
930 CALL PATTERN(#1,PAT):: C
ALL SOUND(1,-5,9):: CALL GCH
AR(X1/8+2,Y1/8+1,PP):: IF PP
<>80 THEN 1080 ELSE 940
940 CALL LOCATE(#1,X1,Y1,#4,
X2,Y2):: CALL PATTERN(#1,PAT
+4):: GOTO 900
950 CALL SOUND(99,220,0,222,
0,224,0)
960 RUND=RUND+1 :: IF YU=1 T
HEN YU=2500 ELSE IF YU=2 THE
N YU=1250
970 SC=SC+YU :: CALL SCO(SC)
:: GOSUB 1090 :: SCHL,AXAX,S
CHW=0 :: CALL HCHAR(1,10,72,
9):: CALL HCHAR(1,20,48+RUND
)
980 IF RUND=4 THEN GOTO 1120
990 CALL FARB(2):: CALL WALD
 :: CALL MITT :: CALL LINK :
: BILD=1 :: CALL LOCATE(#1,1
60,64):: CALL FARB(1):: GOTO
 240
1000 FOR WU=-9 TO 9 :: CALL
MOTION(#4,WU,-8,#1,WU,-4)::
CALL COINC(#4,#2,1+YU,CA)::
CALL COINC(#1,#2,15,CB)
1010 IF CA=-1 THEN 1040 ELSE
 IF CB=-1 THEN 1080
1020 CALL POSITION(#1,X1,Y1)
:: NEXT WU :: CALL MOTION(#1
,0,0,#4,0,0):: CALL COLOR(#4
,8):: CALL LOCATE(#1,72,Y1):
: CALL SOUND(1,-5,0)
1030 GOTO 900
1040 CALL MOTION(#1,0,0,#4,0
,0):: CALL LOCATE(#1,72,Y1):
: FOR A=0 TO 30 :: CALL SOUN
D(-99,-5,A):: NEXT A
1050 CALL POSITION(#2,A1,B1,
#3,A2,B2,#4,X2,Y2):: CALL PA
TTERN(#2,128,#3,136):: CALL
SOUND(1,-5,0):: IF A2>162 TH
EN 1070
1060 CALL MOTION(#2,7,0,#3,7
,0,#4,7,0):: CALL PATTERN(#2
,132,#3,140):: CALL SOUND(1,
-7,0):: GOTO 1050
1070 FOR A=1 TO 15 :: CALL C
OLOR(#2,3,#3,3):: CALL SOUND
(-99,-7,A):: CALL COLOR(#2,4
,#3,4):: NEXT A :: CALL DELS
PRITE(#2,#3,#4):: GOTO 950
1080 CALL MOTION(#1,0,0):: G
OSUB 1090 :: GOTO 350
1090 CALL DELSPRITE(#4,#3,#2
):: CALL CHAR(88,"204098BC99
A9716E7067301E02040804",90,"
",91,"000080403")
1100 CALL CHAR(136,"03030D1E
37634143474E5C180C060204C0C0
B078ECC682C2E2723A1830604020
")
1110 CALL CHAR(140,"00033B7D
26130913070E1C383060408000C0
DCBE64C890C8E070381C0C060201
")):: RETURN
1120 CALL SOU(2):: CALL HCHA
R(11,6,80):: CALL SOU(2):: C
ALL PATTERN(#1,96,#5,96):: C
ALL POSITION(#1,X1,Y1)
1130 CALL LOCATE(#5,X1,Y1-9)
:: CALL COLOR(#5,12):: FOR A
=1 TO 2 :: CALL SOU(1):: CAL
L TEMP(0):: CALL SOU(2):: CA
LL WALD :: NEXT A
1140 CALL SOU(1):: SC=SC+500
0 :: CALL SCO(SC)
1150 CALL ZE(120):: CALL PAT
TERN(#5,112,#1,120):: CALL S
OUND(1,-5,5):: CALL ZE(120):
: CALL PATTERN(#5,96,#1,96):
```

```
: CALL SOUND(1,-7,5)
1160 CALL JOYST(1,X,Y):: IF
X=-4 THEN 1180 ELSE IF X=4 T
HEN 1190 ELSE 1170
1170 FOR A=0 TO 200 STEP 10
:: CALL SOUND(-99,500,8,400+
A,7,600-A,7):: NEXT A :: GOT
O 1150
1180 CALL DELSPRITE(ALL):: C
ALL CLEAR :: STOP
1190 GOTO 210
1200 B=4 :: FOR A=2 TO 24 ST
EP 4 :: CALL POSITION(#1,X1,
Y1):: CALL SOUND(1,-5,A):: C
ALL PATTERN(#1,100)
1210 CALL LOCATE(#1,X1-B,Y1)
:: CALL SOUND(1,-6,A+3):: CA
LL PATTERN(#1,104):: NEXT A
1220 SC=SC+250 :: CALL SCO(S
C):: GOTO 250
1230 CALL COINC(#1,#27,15,CO
):: CALL COINC(#1,#28,15,C1)
:: IF CO OR C1 THEN 1200
1240 GOTO 350
1250 SUB SCO(SC):: CALL HCHA
R(1,2,72,6):: S$=STR$(SC)::
DISPLAY AT(1,2)SIZE(LEN(S$))
BEEP:S$ :: SUBEND
1260 SUB SCHA
1270 X=INT(RND*4)+124
1280 XX=INT(RND*27)+2 :: XY=
INT(RND*9)+8 :: CALL GCHAR(X
Y,XX,PL):: IF PL<>80 THEN 12
80 ELSE CALL HCHAR(XY,XX,X)
1290 FOR A=0 TO 20 STEP 10 :
: CALL SOUND(99,262,A,-1,A+5
,268,A):: NEXT A :: SUBEND
1300 SUB SOU(X):: ON X GOTO
1310,1320
1310 FOR PO=0 TO 30 STEP 5 :
: CALL SOUND(-9,-5,PO,110,PO
):: NEXT PO :: SUBEXIT
1320 FOR PO=30 TO 0 STEP -5
:: CALL SOUND(-9,-7,PO,220,P
O):: NEXT PO :: SUBEND
1330 SUB MONS(X1,X2,RUND)::
MO=5+RUND*2
1340 ON X2 GOTO 1350,1380
1350 CALL CHAR(128,"060F0383
662D1801030707060202040000000
8080C0C0E0E0F2B95FEE70301008
",132,"00060D0306054D3801030
5060603010000000080C2C4E4ECF
8B0C0D8F4400080")
1360 IF X1>140 THEN X2=160 E
LSE IF X1<90 THEN X2=64 ELSE
 X2=112
1370 CALL SPRITE(#2,128,4,X2
,80,0,-MO,#3,136,15,X2-8,160
,0,-MO,#4,128,8,X2+6,240,0,-
MO):: SUBEXIT
```

LISTINGS

```
1380 CALL CHAR(128,"00000101
030307074F9DFA770E0C081060F0
C0C166B41880C0E0E06040202000
",132,"0000000143232737 1F0D0
31B2F020001006 0B0C060A0B21C8
0C0A06060C08000")
1390 IF X1>140 THEN X2=160 E
LSE IF X1<90 THEN X2=64 ELSE
 X2=112
1400 CALL SPRITE(#2,128,8,X2
,20,0,MO,#3,136,12,X2-8,110,
0,MO,#4,128,16,X2+6,185,0,MO
):: SUBEND
1410 SUB MITT :: CALL HCHAR(
8,1,80,512):: CALL HCHAR(12,
1,90,64):: CALL HCHAR(18,1,9
0,64):: CALL HCHAR(24,1,90,3
2):: SUBEND
1420 SUB LINK :: CALL VCHAR(
8,1,87,17):: CALL VCHAR(8,2,
87,17):: FOR A=8 TO 20 STEP
6
1430 DISPLAY AT(A,1)SIZE(4):
"WWWQ" :: DISPLAY AT(A+1,1)S
IZE(3):"WWQ" :: DISPLAY AT(A
+2,1)SIZE(2):"WQ" :: DISPLAY
 AT(A+3,1)SIZE(1):"Q"
1440 NEXT A :: CALL SPRITE(#
27,84,1,89,64):: SUBEND
1450 SUB RECH :: CALL VCHAR(
8,31,87,17):: CALL VCHAR(8,3
2,87,17):: FOR A=8 TO 20 STE
P 6
1460 DISPLAY AT(A,25)SIZE(4)
:"RWWW" :: DISPLAY AT(A+1,26
)SIZE(3):"RWW" :: DISPLAY AT
(A+2,27)SIZE(2):"RW" :: DISP
LAY AT(A+3,28)SIZE(1):"R"
1470 NEXT A :: CALL SPRITE(#
28,84,1,137,184):: SUBEND
1480 SUB CHA1
1490 DATA 33,0F0F0F0CFFFFFFF
F,35,0F0F0F0FFFFFFFC,36,FFFF
FFFF00FFFF00FFFFFF00FFFFFFFF
F0F0F0F03FFFFF03F0F0F030FFFF
FFFF
1500 DATA 40,FFFFFFFF66FFFFD
B7E66425A7E3C3C1818183C3C3C7
E7EFFFFFFFFFF,44,7FFFFF7F,45
,FEFFFFFE
1510 DATA 64,0F0D0D0D0D1B367
D507070F0B0B8DCEE1D35351D0F1
F1D0F0B1B1B0A0A0E0E0B,68,F05
0D85858D8F8F0B0A0E0F0D8D8D85
0EFAFBEBCBCBEACE8D7F63E3E1A1
B3D35
1520 DATA 72,FFFFFFFFFFFFFFF
F,73,FFFF,74,FFFFFFFF,75,FFF
FFFFFFFFF0F0F,76,FFFFFFFFFFF
FF0F,77,0000101010 1,78,00000
0101,79,0010101010101
```

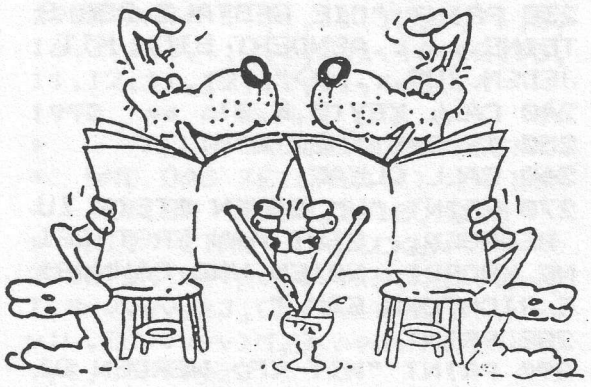
```
1530 DATA 80,FFFFFFFFFFFFFFF
F0103070F1F3F7FFF80C0E0F0F8F
CFEFF,84,427E42427E42427E424
27E42427E4242,86,,87,,90,
1540 DATA 88,204098BC99A9716
E7067301E02040804,90,,91,000
080403,92,1824425A5A3C243C18
3C242424242442,94,,95,
1550 DATA 63,7E2424242424247
E,96,183C181866FFBDBD99BDBD2
424242442,98,,99,,100,183C18
1876FFB9B798BCBC2C664C4884,1
02,,103,
1560 DATA 104,183C18186EFF9D
ED193D3D7446221221,106,,107,
,108,00060F062A5C98BBB018142
24C48804,110,,111,
1570 DATA 112,00060F060A1C14
34321914F282020201,114,,115,
,116,0060F060543A19DD0D18284
432120102,118,,119,
1580 DATA 120,0060F060503828
2C4C98284F4140408,122,,123,,
124,000040A0BFA545,125,10103
8101010101,126,1F2345FDB5858
6FC,127,7F30FC783
1590 DATA 136,03030D1E376341
43474E5C180C060204C0C0B078EC
C682C2E2723A1830604020,140,0
0033B7D26130913070E1C3830604
08000C0DCBE64C890C8E070381C0
C060201
1600 DATA 34,000000003C7EE7E
7,0,
1610 RESTORE 1490
1620 READ A,A$ :: IF A=0 THE
N 1630 ELSE CALL CHAR(A,A$):
: GOTO 1620
1630 SUBEND
1640 SUB FARB(X):: ON X GOTO
 1650,1670
1650 CALL SCREEN(13):: CALL
COLOR(1,15,4,2,15,4,3,16,1,4
,16,1,5,7,4,6,13,4,7,5,2,8,7
,7,12,8,5)
1660 CALL COLOR(#1,16,#27,11
,#28,11):: SUBEXIT
1670 CALL DELSPRITE(#27,#28,
#5,#2,#3,#4):: CALL COLOR(#1
,1):: CALL SCREEN(2):: CALL
COLOR(5,1,1,6,1,1,7,1,1,8,1,
1)
1680 CALL COLOR(12,1,1,1,1,1
,2,1,1):: SUBEND
1690 SUB WALD
1700 DISPLAY AT(3,1):"GFIJKL
JIGFJKLJGFJKLJGFJKLJGFBD M O
 NCE NNOBEMONMCDOON CECEO N
M BDMO NCDON OBEMN OBDBD NO.
.MCE NO BE MONCDM O CE@ANM O
N @AO..N@AMNO @AONMO@A"
```

LISTINGS

```
1710 SUBEND
1720 SUB ZE(X):: FOR X1=1 TO
 X :: NEXT X1 :: SUBEND
1730 SUB TEMP(X)
1740 DISPLAY AT(3,12)SIZE(8)
:",(++++(-" :: DISPLAY AT(4,
13)SIZE(6):")MONO)" :: DISPL
AY AT(5,13)SIZE(6):"*ON M*"
1750 DISPLAY AT(6,12)SIZE(8)
:"########&" :: DISPLAY AT(7,
11)SIZE(10):"!%%%%%%%%'"
1760 CALL HCHAR(5,17,34):: I
F X=1 THEN CALL SPRITE(#5,92
,2,20,136):: CALL HCHAR(5,17
,79)
1770 SUBEND
```

LET'S JUMP

Let's jump ist ein Brettspiel für die TI-Konsole oder für Ext.Basic. Ziel des Spieles ist es, alle Steine des Gegners zu schlagen oder mehr als 16 Punkte zu erreichen. Jeder Spieler erhält 8 Steine, die in römischen Ziffern ihre jeweilige Wertigkeit anzeigen. Diese ändert sich von Zug zu Zug, und kann einen Wert von 1 – 4 annehmen.
Diese Wertigkeit beeinflußt die Zugmöglichkeiten. Mit einem Stein der Wertigkeit II kann man z.B. genau drei Felder senkrecht oder waagerecht springen. Die Felder dazwischen werden nicht berührt.
Für einen geschlagenen Stein erhält man einen Treffer (T) und soviel Punkte (P), wie der geschlagene Stein anzeigte. Die Treffer und Punkte werden jeweils unter den Namen angezeigt.
Nach dem Programmstart wird man gefragt, ob man gegen den Computer spielen will. Beantwortet man diese Frage mit 'J', so wird man danach nach der Schwierigkeitsstufe gefragt. 1 ist die einfachste und 3 die schwerste Stufe.
Ein Tip: Der Computer schlägt auf allen Stufen auch gedeckte Steine.



LISTINGS

```
10 REM**********************
11 REM*    LET'S JUMP      *
12 REM*                    *
13 REM*   Copyright by     *
14 REM*                    *
15 REM* Oliver Wallenfang* 
16 REM*                    *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole  *
19 REM*                    *
23 REM* Speicherbelegung  *
24 REM*    9333 Bytes      *
25 REM**********************
26 REM
100 CALL SCREEN(3)
110 RESTORE
120 CALL CLEAR
130 PRINT TAB(9);"LET'S JUMP
"
140 PRINT
150 PRINT "ZIEL DIESES BRETT
SPIELES ISTES, ENTWEDER ALLE
 STEINE DESGEGNERS ZU SCHLAG
EN,"
160 PRINT "ODER MEHR ALS 16
PUNKTE ZU..ERREICHEN, WODURC
H DAS SPIELEVENTUELL ABGEKUE
RZT WIRD."
170 PRINT
180 PRINT "DIE SPIELSTEINE H
ABEN DIE...GESTALT VON ROEMI
SCHEN......ZAHLEN. DIESE ZAH
LEN ZEIGEN,"
190 PRINT "WIEVIELE FELDER M
AN MIT DEM JEWEILIGEN STEIN
HORIZONTAL ODER VERTIKAL SPR
INGEN KANN."
200 PRINT
210 PRINT "ES IST NICHT MOEG
LICH, MEHR ODER WENIGER FELD
ER ZU......SPRINGEN, ALS ANG
EZEIGT."
220 PRINT
230 PRINT "DIE GESTALT DER S
TEINE......AENDERT SICH MIT
JEDEM ZUG. ..>>"
240 CALL KEY(0,K,S)
250 IF S=0 THEN 240
260 CALL CLEAR
270 PRINT "UM EINEN STEIN ZU
 BEWEGEN,..GIBT MAN ERST SEI
NE KOORDI- NATEN UND DANN DI
E RICHTUNG EIN."
280 PRINT
290 PRINT "MIT <F> WERDEN DI
E EINGABEN GELOESCHT."
300 PRINT
310 PRINT "DAS PROGRAMM IST
IN TI-BASICGESCHRIEBEN, DA A
BER KEINE  ZEICHEN UEBER 144
 BENUTZT"
320 PRINT "WURDEN, LAUFT ES
OHNE AEN-..DERUNG AUCH IN EX
-BASIC,....WAS DAS SPIEL ERH
EBLICH.....VERSCHNELLERT."
330 PRINT
340 INPUT "GEGEN DEN COMPUTE
R ? (J/N)..":AN$
350 COMP=-(AN$="J")-(AN$="j"
)
360 IF COMP=0 THEN 400
370 CALL CLEAR
380 INPUT "SPIELSTAERKE ? (1
-3).......":SPI
390 IF (SPI<1)+(SPI>3)+(SPI<
>INT(SPI))THEN 370
400 RANDOMIZE
410 CALL CLEAR
420 NAME$(2)="TI-99"
430 FOR I=1 TO 2
440 CALL CLEAR
450 PRINT "NAME VON SPIELER"
;I:">>> ";
460 INPUT "":NAME$(I)
470 IF LEN(NAME$(I))>8 THEN
440
480 PRINT : :"RICHTIG ? J/N"
490 CALL KEY(0,K,S)
500 IF S<>1 THEN 490
510 IF K=78 THEN 420
520 IF COMP=1 THEN 540
530 NEXT I
540 FOR I=1 TO 2
550 PUN(I)=0
560 TREFF(I)=0
570 NEXT I
580 CALL CLEAR
590 DIM WERT(8,8)
600 IF DURCH=1 THEN 830
610 DURCH=1
620 DATA 3810101010101038,E7
4242424242E7,FF92929292929
2FF,FF52524A4A4646FF
630 RESTORE
640 FOR I=96 TO 99
650 READ A$
660 CALL CHAR(I,A$)
670 CALL CHAR(I+8,A$)
680 CALL CHAR(I+16,A$)
690 CALL CHAR(I+24,A$)
700 NEXT I
710 CALL CHAR(59,"081C2A0808
080808")
720 CALL CHAR(60,"002040FF40
20")
730 CALL CHAR(61,"0808080808
2A1C08")
740 CALL CHAR(62,"000402FF02
04")
750 CALL CHAR(128,"")
760 CALL CHAR(136,"")
770 CALL COLOR(9,16,11)
```

```
780 CALL COLOR(10,16,7)
790 CALL COLOR(11,2,11)
800 CALL COLOR(12,2,7)
810 CALL COLOR(13,11,11)
820 CALL COLOR(14,7,7)
830 CALL COLOR(13,1,1)
840 CALL COLOR(14,1,1)
850 A$=CHR$(128)&CHR$(128)&C
HR$(128)
860 B$=CHR$(136)&CHR$(136)&C
HR$(136)
870 C$=A$&B$&A$&B$&A$&B$&A$&
B$&A$
880 FOR I=1 TO 4
890 FOR I1=1 TO 3
900 PRINT SEG$(C$,3,22);
910 NEXT I1
920 FOR I1=1 TO 3
930 PRINT SEG$(C$,6,22);
940 NEXT I1
950 NEXT I
960 CALL VCHAR(1,1,128,48)
970 FOR I=4 TO 24 STEP 6
980 CALL VCHAR(I,1,136,3)
990 CALL VCHAR(I,2,136,3)
1000 NEXT I
1010 CALL COLOR(13,11,11)
1020 CALL COLOR(14,7,7)
1030 FOR I=1 TO 4
1040 FOR J=1 TO 8 STEP 7
1050 WERT(J,I)=I
1060 WERT(J,9-I)=I
1070 NEXT J
1080 NEXT I
1090 FOR I=1 TO 8 STEP 7
1100 FOR I1=1 TO 8
1110 SP=1-(I=8)
1120 HFIG(SP,I1)=I
1130 BFIG(SP,I1)=I1
1140 CALL GCHAR(I*3,I1*3,ZX)
1150 CALL HCHAR(I*3-1,I1*3-1
,ZX-33+WERT(I,I1)-((I=8)*16)
)
1160 NEXT I1
1170 NEXT I
1180 SP=2
1190 A$="0125"&NAME$(1)
1200 GOSUB 2800
1210 A$="02250P 0T"
1220 GOSUB 2800
1230 A$="0425"&NAME$(2)
1240 GOSUB 2800
1250 A$="05250P 0T"
1260 GOSUB 2800
1270 A$="0725HOEHE"
1280 GOSUB 2800
1290 A$="1125BREITE"
1300 GOSUB 2800
1310 A$="1625; = > <"
1320 GOSUB 2800
1330 A$="17251 2 3 4"
1340 GOSUB 2800
1350 SP=3-SP
1360 HZUG=0
1370 BZUG=0
1380 IF (K=70)+(K=102)=0 THE
N 1420
1390 CALL HCHAR(HO*3-1,24,ZX
1)
1400 IF BR=0 THEN 1420
1410 CALL HCHAR(1,BR*3-1,ZX2
)
1420 REM
1430 CALL HCHAR(9,25,32,8)
1440 CALL HCHAR(13,25,32,8)
1450 CALL HCHAR(19,25,32,8)
1460 FOR I=21 TO 24
1470 CALL HCHAR(I,25,32,8)
1480 NEXT I
1490 IF (SP=2)*(COMP=1)THEN
2880
1500 CALL KEY(0,K,S)
1510 IF S<>1 THEN 1500
1520 HO=K-48
1530 IF HO<1 THEN 1500
1540 IF HO>8 THEN 1500
1550 CALL HCHAR(9,25,K)
1560 CALL GCHAR(HO*3-1,24,ZX
1)
1570 CALL HCHAR(HO*3-1,24,60
)
1580 CALL KEY(0,K,S)
1590 IF S<>1 THEN 1580
1600 IF (K=70)+(K=102)THEN 1
390
1610 BR=K-48
1620 IF BR<1 THEN 1580
1630 IF BR>8 THEN 1580
1640 CALL HCHAR(13,25,K)
1650 CALL GCHAR(1,BR*3-1,ZX2
)
1660 CALL HCHAR(1,BR*3-1,61)
1670 CALL KEY(0,K,S)
1680 IF S<>1 THEN 1670
1690 IF (K=70)+(K=102)THEN 1
390
1700 RI=K-48
1710 IF RI<1 THEN 1670
1720 IF RI>4 THEN 1670
1730 CALL HCHAR(19,25,K)
1740 CALL KEY(0,K,S)
1750 IF S<>1 THEN 1740
1760 IF (K=70)+(K=102)THEN 1
390
1770 REM ZUGBERECHNUNG
1780 ST=0
1790 CALL HCHAR(HO*3-1,24,ZX
1)
1800 CALL HCHAR(1,BR*3-1,ZX2
)
1810 FOR I=1 TO 8
1820 IF HFIG(SP,I)<>HO THEN
```

LISTINGS

LISTINGS

```
1830 IF BFIG(SP,I)<>BR THEN
1850
1840 ST=I
1850 NEXT I
1860 IF ST=0 THEN 2660
1870 HS=HFIG(SP,ST)
1880 BS=BFIG(SP,ST)
1890 HZ=0
1900 BZ=0
1910 IF RI<>1 THEN 1930
1920 HZ=-WERT(HS,BS)
1930 IF RI<>2 THEN 1950
1940 HZ=WERT(HS,BS)
1950 IF RI<>3 THEN 1970
1960 BZ=WERT(HS,BS)
1970 IF RI<>4 THEN 1990
1980 BZ=-WERT(HS,BS)
1990 HO=HS+HZ
2000 BR=BS+BZ
2010 IF HO<1 THEN 2660
2020 IF HO>8 THEN 2660
2030 IF BR<1 THEN 2660
2040 IF BR>8 THEN 2660
2050 CALL GCHAR(HO*3-1,BR*3-
1,ZX)
2060 IF ZX>127 THEN 2100
2070 IF (SP=1)*(ZX<112)THEN
2660
2080 IF (SP=2)*(ZX>111)THEN
2660
2090 GOTO 2380
2100 HF=HFIG(SP,ST)*3
2110 BF=BFIG(SP,ST)*3
2120 CALL GCHAR(HF,BF,ZX1)
2130 CALL GCHAR(HO*3-1,BR*3-
1,ZX2)
2140 CALL GCHAR(HF-1,BF-1,ZX
3)
2150 WERT(HO,BR)=INT(4*RND)+
1
2160 ZE=ZX-33+WERT(HO,BR)-((
SP=2)*16)
2170 FOR I=1 TO 4
2180 CALL HCHAR(HF-1,BF-1,ZX
3)
2190 CALL HCHAR(HO*3-1,BR*3-
1,ZX2)
2200 CALL SOUND(200,33000,30
)
2210 CALL SOUND(10,440,0)
2220 CALL HCHAR(HF-1,BF-1,ZX
1)
2230 CALL HCHAR(HO*3-1,BR*3-
1,ZE)
2240 CALL SOUND(200,33000,30
)
2250 CALL SOUND(10,660,0)
2260 NEXT I
2270 HFIG(SP,ST)=HO
2280 BFIG(SP,ST)=BR
2290 HZ=0
2300 BZ=0
2310 BR=0
2320 HO=0
2330 WEI=0
2340 RI=0
2350 IF TREFF(SP)>7 THEN 255
0
2360 IF PUN(SP)>16 THEN 2550
2370 GOTO 1350
2380 GSP=3-SP
2390 TR=0
2400 FOR I=1 TO 8
2410 IF HFIG(GSP,I)<>HO THEN
 2520
2420 IF BFIG(GSP,I)<>BR THEN
 2520
2430 TR=1
2440 CALL SOUND(200,440,0)
2450 CALL SOUND(300,220,0)
2460 CALL SOUND(500,-6,0)
2470 TREFF(SP)=TREFF(SP)+1
2480 PUN(SP)=PUN(SP)+WERT(HF
IG(GSP,I),BFIG(GSP,I))
2490 A$="0"&STR$((SP*3)-1)&"
25"&STR$(PUN(SP))&"P "&STR$(
TREFF(SP))&"T "
2500 GOSUB 2800
2510 HFIG(GSP,I)=0
2520 NEXT I
2530 CALL GCHAR(HO*3,BR*3,ZX
)
2540 GOTO 2100
2550 A$="1203"&NAME$(SP)&" H
AT GEWONNEN !"
2560 GOSUB 2800
2570 A$="1504NOCH EINMAL ? (
J/N)"
2580 GOSUB 2800
2590 CALL KEY(0,K,S)
2600 IF S<>1 THEN 2590
2610 IF K=ASC("J")THEN 100
2620 IF K=ASC("j")THEN 100
2630 CALL CLEAR
2640 PRINT TAB(9);"SCHADE !"
: : : : : : : : : :
2650 END
2660 REM FALSCH !
2670 A$="2125NICHT"
2680 GOSUB 2800
2690 A$="2225-----"
2700 GOSUB 2800
2710 A$="2325MOEGLICH"
2720 GOSUB 2800
2730 A$="2425--------"
2740 GOSUB 2800
2750 CALL SOUND(1000,110,0)
2760 CALL SOUND(10,110,5)
2770 HFIG(SP,ST)=HO-HZ
2780 BFIG(SP,ST)=BR-BZ
2790 GOTO 1360
2800 REM SCHRIFT
```

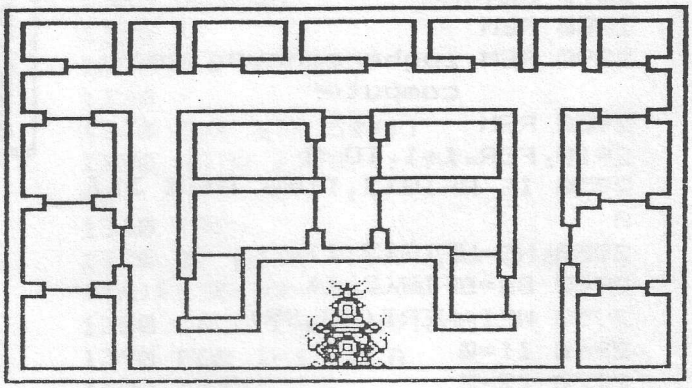
```
2810 H=VAL(SEG$(A$,1,2))
2820 B=VAL(SEG$(A$,3,2))
2830 FOR II=5 TO LEN(A$)
2840 ZEI=ASC(SEG$(A$,II,1))
2850 CALL HCHAR(H,B+II-5,ZEI
)
2860 NEXT II
2870 RETURN
2880 REM
2890 REM zugberechnung vom
        computer
2900 REM
2910 FOR I=1 TO 8
2920 IF HFIG(2,I)=0 THEN 317
0
2930 HO=HFIG(2,I)
2940 BR=BFIG(2,I)
2950 WEI=WERT(HO,BR)
2960 Z1=0
2970 Z2=0
2980 Z3=0
2990 Z4=0
3000 IF HO-WEI<1 THEN 3020
3010 CALL GCHAR((HO-WEI)*3-1
,BR*3-1,Z1)
3020 IF HO+WEI>8 THEN 3040
3030 CALL GCHAR((HO+WEI)*3-1
,BR*3-1,Z2)
3040 IF BR+WEI>8 THEN 3060
3050 CALL GCHAR(HO*3-1,(BR+W
EI)*3-1,Z3)
3060 IF BR-WEI<1 THEN 3080
3070 CALL GCHAR(HO*3-1,(BR-W
EI)*3-1,Z4)
3080 RI=1
3090 IF (Z1<112)*(Z1>95)THEN
 4000
3100 RI=2
3110 IF (Z2<112)*(Z2>95)THEN
 4000
3120 RI=3
3130 IF (Z3<112)*(Z3>95)THEN
 4000
3140 RI=4
3150 IF (Z4<112)*(Z4>95)THEN
 4000
3160 CALL SOUND(10,880,0)
3170 NEXT I
3180 REM
3190 REM defensiv spielen
3200 REM
3210 FOR I=1 TO 8
3220 IF SPI=1 THEN 3450
3230 IF HFIG(2,I)=0 THEN 345
0
3240 HO=HFIG(2,I)
3250 BR=BFIG(2,I)
3260 GOSUB 4070
3270 IF FE=0 THEN 3450
3280 WEI=WERT(HO,BR)
3290 HO1=HO
```

LISTINGS

```
3300 BR1=BR
3310 FOR RI=1 TO 4
3320 RH=(RI=1)-(RI=2)
3330 RB=(RI=4)-(RI=3)
3340 HO=HO+(WEI*RH)
3350 BR=BR+(WEI*RB)
3360 IF (HO<1)+(HO>8)THEN 34
20
3370 IF (BR<1)+(BR>8)THEN 34
20
3380 CALL GCHAR(HO*3-1,BR*3-
1,ZX)
3390 IF ZX<128 THEN 3420
3400 GOSUB 4070
3410 IF FE=0 THEN 3980
3420 HO=HO1
3430 BR=BR1
3440 NEXT RI
3450 NEXT I
3460 CALL SOUND(100,440,0)
3470 ZU=0
3480 REM
3490 REM verlegenheitszug
3500 REM
3510 ZZ=INT(8*RND)+1
3520 IF HFIG(2,ZZ)=0 THEN 35
10
3530 ZU=ZU+1
3540 HO=HFIG(2,ZZ)
3550 BR=BFIG(2,ZZ)
3560 WEI=WERT(HO,BR)
3570 I=ZZ
3580 R=INT(4*RND)+1
3590 FOR RIC=R TO R+4
3600 RI=RIC+((RIC>4)*4)
3610 ON RI GOTO 3620,3650,37
10,3680
3620 IF HO-WEI<1 THEN 3740
3630 CALL GCHAR((HO-WEI)*3-1
,BR*3-1,ZX)
3640 IF ZX>123 THEN 3770 ELS
E 3740
3650 IF HO+WEI>8 THEN 3740
3660 CALL GCHAR((HO+WEI)*3-1
,BR*3-1,ZX)
3670 IF ZX>123 THEN 3770 ELS
E 3740
3680 IF BR-WEI<1 THEN 3740
3690 CALL GCHAR(HO*3-1,(BR-W
EI)*3-1,ZX)
3700 IF ZX>123 THEN 3770 ELS
E 3740
3710 IF BR+WEI>8 THEN 3740
3720 CALL GCHAR(HO*3-1,(BR+W
EI)*3-1,ZX)
3730 IF ZX>123 THEN 3770
3740 NEXT RIC
3750 IF ZU>9 THEN 3960
3760 GOTO 3480
3770 IF SPI<3 THEN 4000
3780 IF ZU>9 THEN 4000
```

```
3790 WEI=WERT(HO,BR)
3800 HC1=HO
3810 BC1=BR
3820 RH=(RI=1)-(RI=2)
3830 RB=(RI=4)-(RI=3)
3840 HC2=HO
3850 HO=HO+(WEI*RH)
3860 IF HO<>HC2 THEN 3880
3870 BR=BR+(WEI*RB)
3880 GOSUB 4070
3890 HO=HC1
3900 BR=BC1
3910 IF FE=0 THEN 4000
3920 IF (FE=-1)*(ZU<10)THEN
3480
3930 FOR ZZ=1 TO 8
3940 IF HFIG(2,ZZ)=0 THEN 39
60
3950 GOTO 3530
3960 NEXT ZZ
3970 GOTO 4000
3980 HO=HO1
3990 BR=BR1
4000 REM
4010 REM stein setzen
4020 REM
4030 ST=I
4040 HO=HFIG(2,ST)
4050 BR=BFIG(2,ST)
4060 GOTO 1870
4070 REM feld pruefen
4080 HP=HO
4090 BP=BR
4100 FOR J=1 TO 8
4110 IF HFIG(1,J)=0 THEN 420
0
4120 HF=HFIG(1,J)
4130 BF=BFIG(1,J)
4140 WF=WERT(HF,BF)
4150 IF (HF-WF=HP)*(BF=BP)TH
EN 4230
4160 IF (HF+WF=HP)*(BF=BP)TH
EN 4230
4170 IF (HF=HP)*(BF+WF=BP)TH
EN 4230
4180 IF (HF=HP)*(BF-WF=BP)TH
EN 4230
4190 FE=0
4200 NEXT J
4210 CALL SOUND(10,660,0)
4220 RETURN
4230 CALL SOUND(100,220,0)
4240 FE=-1
4250 RETURN
```

E.T.



Helfen Sie E.T. seine Maschine zusammenzubauen, damit er nach Hause telefonieren kann.
Falls Sie einen Sprachsynthesizer haben, gibt es schon beim Titelbild eine kleine Überraschung. Drükken Sie es nicht zu schnell weg!
Zuerst wird Ihnen die Maschine, die zusammengebaut werden muß, gezeigt. Prägen Sie sich sie gut ein! Die einzelnen Teile der Maschine werden nun in die verschiedenen Räume verteilt und E.T. erscheint auf dem Bildschirm. In drei Räumen sind keine Maschinenteile, sondern ein Schlüssel, eine Bierdose und Kekse. Alle Türen sind verschlossen.
E.T. hat zu Anfang einen Schlüssel bei sich. Vor dem Öffnen einer Tür muß sich E.T. jeweils einen Schlüssel im „Schlüssel"-Raum holen. Gehen Sie daher zuerst zu diesem Raum, um diese Tür zu öffnen. Jede Tür, die einmal geöffnet wurde, bleibt offen! Jeder Schlüssel kann nur einmal verwendet werden. Wenn E.T. seine Farbe ändert, so benötigt er Nahrung (blau) oder hat Durst (gelb). Gehen Sie dann in den entsprechenden Raum, um E.T. seinen Wunsch zu erfüllen (beim jeweils ersten Mal Schlüssel zum Öffnen der Tür nicht vergessen), sonst stirbt E.T. nach einiger Zeit.
Im oberen Gang läuft ein Hund (sehr gute Animation). Wenn er E.T. beißt, so stirbt dieser, da er als Außerirdischer nicht über die nötigen Abwehrstoffe verfügt.
Die Maschine besteht aus 16 Teilen. Um ein Teil aufzunehmen, führen Sie E.T. über das Teil. Um es abzulegen, drücken Sie die Feuertaste. Das Teil erscheint dann immer an E.T.'s rechtem Fuß!!!
Bauen Sie also die Maschine von rechts nach links!! Falls Sie wegen eines Fehlers die Maschine abbauen müssen, so hat dies am besten von links nach rechts zu geschehen! Falls Sie in der Mitte anfangen würden, könnte alles um E.T. herum gelöscht werden und das Spiel verloren sein. E.T. kann immer nur ein Teil tragen. Falls Sie der Meinung sind, daß die Maschine richtig zusammengebaut ist, so drücken Sie die Feuertaste und danach die 'SPACE'-Taste der Konsole.
Es gibt keine zeitliche Begrenzung für das Spiel.

Punktebewertung:
Kekse essen: 150 Punkte
Bier trinken: 150 Punkte
Schlüssel holen: 200 Punkte
Fertige Maschine: Bonus
Oben links im Bild wird der Punktestand angezeigt, oben rechts das Teil, das E.T. gerade trägt.

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *         E.T.         *
13 ! *                      *
14 ! *     Copyright by     *
15 ! *                      *
16 ! *  Manfred Lipowski    *
17 ! *                      *
18 ! * Benoetigte Geraete   *
19 ! *  TI99/4A  Konsole    *
20 ! *      Ext. Basic      *
21 ! *      Joystick 1      *
22 ! * opt. Sprachsynth.    *
23 ! *       Cassette       *
24 ! *(oder Disk+32k-Erw.)  *
25 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     11880 Bytes      *
28 ! *                      *
29 ! ************************
100 CALL CLEAR :: CALL CHARS
ET :: RANDOMIZE :: CALL SCRE
EN(2):: CALL MAGNIFY(3):: CA
LL CHARPAT(88,A$,89,B$,90,C$
):: CALL CHAR(63,C$,62,B$,61
,A$):: CALL Q
110 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL SCREEN(5)::
 FOR A=1 TO 14 :: CALL COLOR
(A,12,1):: NEXT A
120 CALL W :: CALL COLOR(5,1
6,7,6,16,7,7,15,1):: CALL Z
:: CALL P
130 CALL SPRITE(#3,36,8,24,8
9,#4,60,15,26,116,#5,40,11,2
5,163,#6,88,2,48,120)
140 CALL SPRITE(#1,104,4,153
,121):: CALL H(PAT):: AS,JA,
BE,G=0 :: KE=1
150 CALL H(PAT):: AS,JA,BE,G
=0 :: KE=1 :: CALL F(330)::
CALL F(262):: CALL F(294)::
CALL F(196)
160 CALL PATTERN(#1,104,#6,P
AT)
170 CALL JOYST(1,X,Y):: CALL
 POSITION(#1,X1,Y1):: IF RND
<.1 THEN CALL H(PAT)
180 CALL KEY(1,K,S):: IF K=1
8 THEN 410
190 CALL PATTERN(#6,PAT+4)::
 CALL COINC(ALL,ZA):: IF ZA
THEN 720
200 IF JA>125 THEN 210 ELSE
IF BE>250 THEN 220 ELSE 230
210 CALL COLOR(#1,8):: GOTO
230
220 CALL COLOR(#1,12)
230 IF X=-4 THEN 240 ELSE IF
 X=4 THEN 250 ELSE IF Y=-4 T
HEN 260 ELSE IF Y=4 THEN 270
 ELSE IF X=0 AND Y=0 THEN 16
0
240 Y1=Y1-8 :: PA=128 :: TON
=-5 :: GOTO 280
250 Y1=Y1+8 :: PA=136 :: TON
=-6 :: GOTO 280
260 X1=X1+8 :: PA=108 :: TON
=-7 :: GOTO 280
270 X1=X1-8 :: PA=108 :: TON
=-7.35
280 CALL PATTERN(#1,PA,#6,PA
T):: CALL SOUND(-9,TON,9)
290 CALL GCHAR(X1/8+2,Y1/8+1
,XX):: CALL GCHAR(X1/8+1,Y1/
8+1,YY):: CALL GCHAR(X1/8+1,
Y1/8+2,ZZ):: CALL GCHAR(X1/8
+2,Y1/8+2,WW)
300 IF XX<>32 OR YY<>32 OR Z
Z<>32 OR WW<>32 THEN 310 ELS
E 350
310 IF XX>=64 AND XX<=77 OR
YY>=64 AND YY<=77 OR ZZ>=64
AND ZZ<=77 OR WW>=64 AND WW<
=77 THEN 770
320 IF XX=80 OR XX=81 OR YY=
80 OR YY=81 OR ZZ=80 OR ZZ=8
1 OR WW=80 OR WW=81 THEN 330
 ELSE 340
330 IF KE<>1 THEN 770 ELSE C
ALL VCHAR(X1/8+1,Y1/8+1,32,2
):: CALL VCHAR(X1/8+1,Y1/8+2
,32,2):: KE=0 :: CALL SOUND(
-99,-2,9):: GOTO 350
340 IF XX>=116 AND XX<=127 O
R XX>=44 AND XX<=47 OR YY>=1
16 AND YY<=127 OR YY>=44 OR
YY<=47 OR ZZ>=116 AND ZZ<=12
7 OR ZZ>=44 AND ZZ<=47 OR WW
>=116 AND WW<=127 OR WW>=44
AND WW<=47 THEN 370
350 CALL COINC(ALL,ZA):: IF
ZA THEN 720
360 CALL LOCATE(#1,X1,Y1)::
CALL PATTERN(#1,PA+4,#6,PAT+
4):: CALL SOUND(-9,TON,4)::
JA=JA+1 :: BE=BE+1 :: IF JA>
=350 OR BE>=400 THEN 770 ELS
E 170
370 IF AS=0 THEN 380 ELSE CA
LL SOUND(999,-3,0):: CALL T(
333):: GOTO 350
380 IF XX<>32 THEN AS=XX ELS
E IF YY<>32 THEN AS=YY ELSE
IF ZZ<>32 THEN AS=ZZ ELSE IF
 WW<>32 THEN AS=WW
390 CALL VCHAR(X1/8+1,Y1/8+1
,32,2):: CALL VCHAR(X1/8+1,Y
1/8+2,32,2):: FOR A=0 TO 30
STEP 2.5 :: CALL SOUND(-99,5
20,A,522,A):: NEXT A
400 CALL HCHAR(1,28,AS):: GO
```

```
TO 350
410 CALL GCHAR(X1/8+2,Y1/8+2
,KK):: IF KK=32 THEN 420 ELS
E CALL F(222):: CALL T(999):
: GOTO 170
420 IF AS=0 THEN 430 ELSE CA
LL GCHAR(X1/8+2,Y1/8+1,KK)::
 IF KK<>32 THEN 430 ELSE CAL
L HCHAR(1,28,32,4):: CALL HC
HAR(X1/8+2,Y1/8+2,AS):: AS=0
430 CALL KEY(0,K,S):: IF K=3
2 THEN 440 ELSE 170
440 CALL GCHAR(23,15,AZ):: I
F AZ<>117 THEN 170
450 CALL GCHAR(23,16,AZ):: I
F AZ<>119 THEN 170
460 CALL GCHAR(23,17,AZ):: I
F AZ<>121 THEN 170
470 CALL GCHAR(23,18,AZ):: I
F AZ<>123 THEN 170
480 CALL F(196):: CALL GCHAR
(22,15,AZ):: IF AZ<>116 THEN
 170
490 CALL GCHAR(22,16,AZ):: I
F AZ<>118 THEN 170
500 CALL GCHAR(22,17,AZ):: I
F AZ<>120 THEN 170
510 CALL GCHAR(22,18,AZ):: I
F AZ<>122 THEN 170
520 CALL F(247):: CALL GCHAR
(21,15,AZ):: IF AZ<>124 THEN
 170
530 CALL GCHAR(21,16,AZ):: I
F AZ<>126 THEN 170
540 CALL GCHAR(21,17,AZ):: I
F AZ<>44 THEN 170
550 CALL GCHAR(21,18,AZ):: I
F AZ<>45 THEN 170
560 CALL F(294):: CALL GCHAR
(20,16,AZ):: IF AZ<>125 THEN
 170
570 CALL GCHAR(20,17,AZ):: I
F AZ<>127 THEN 170
580 CALL F(392):: CALL GCHAR
(19,16,AZ):: IF AZ<>46 THEN
170
590 CALL GCHAR(19,17,AZ):: I
F AZ<>47 THEN 170
600 CALL F(392):: CALL T(999
):: CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL SCREEN(2)
610 CALL CHAR(80,"2020E01008
09091F1E0E0E06060303001F7040
C0808708F03CFF7FFF3FFFFFFF",
84,"F80E020301E1100F3CFFFFFF
FFFFFFFF040407081090 90F81870
706060C0C000")
620 CALL CHAR(88,"01",89,"",
90,"",91,""):: FOR A=4 TO 27
 :: CALL SPRITE(#A,88,16,INT
(RND*200)+20,INT(RND*240)+10
```

LISTINGS

```
):: NEXT A
630 CALL SPRITE(#1,80,15,120
,110,#2,84,15,120,126,#3,104
,4,120,118):: CALL SAY("E+T
GO+HOME"):: FOR A=4 TO 28 ::
 CALL MOTION(#A,INT(RND*25)+
8,0)
640 NEXT A
650 DATA 5,330,5,392,5,392,2
.5,349,2.5,330,6,294,5,349,5
,440,5,440,7,392
660 FOR D=1 TO 3 :: RESTORE
650 :: FOR A=1 TO 10 :: READ
 C,B :: CALL SOUND(C*100,B,0
,B+2,0):: NEXT A :: NEXT D
670 FOR A=1 TO 45 :: G=G+50
:: CALL K(G):: CALL SOUND(1,
-1,1):: NEXT A
680 CALL F(196):: CALL F(196
):: CALL F(196):: CALL SAY("
PRESS+ANY KEY+TO START OR+SP
ACE+TO STOP")
690 CALL KEY(0,K,S):: IF S=0
 THEN 690 ELSE IF K=32 THEN
700 ELSE 710
700 CALL F(112):: CALL F(220
):: CALL F(112):: CALL DELSP
RITE(ALL):: CALL CLEAR :: CA
LL SAY("BYE+BYE"):: STOP
710 CALL F(440):: CALL F(349
):: CALL F(392):: CALL F(262
):: CALL DELSPRITE(ALL):: CA
LL SAY("O+K LET+GO"):: GOTO
100
720 CALL COINC(#1,#6,10,BC):
: IF BC THEN 770
730 CALL COINC(#1,#3,20,CC):
: IF CC THEN 840
740 CALL COINC(#1,#4,20,DC):
: IF DC THEN 890
750 CALL COINC(#1,#5,20,EC):
: IF EC THEN 930
760 GOTO 170
770 CALL MOTION(#6,0,0):: FO
R A=0 TO 30 STEP 2 :: CALL S
OUND(-99,-5,A):: NEXT A :: C
ALL SAY("UHOH"):: FOR A=1 TO
 14 :: B=INT(RND*13)+3 :: IF
 B=5 THEN B=6
780 C=INT(RND*7)+1 :: CALL S
OUND(-99,-C,A*2):: CALL COLO
R(#1,B):: NEXT A
790 FOR A=25 TO 5 STEP -5 ::
 CALL SOUND(-99,-5,A):: CALL
 SOUND(-99,-5,2+A):: NEXT A
:: CALL DELSPRITE(#1)
800 DATA 8,196,3,262,3,294,3
,330,6,294,9,247,8,196,3,262
,3,294,4,330,9,294
810 DATA 8,196,3,262,3,294,3
,330,6,294,9,247,3,247,3,247
```

```
,3,262,4,247,6,196,6,196
820 RESTORE 800 :: FOR A=1 T
O 23 :: READ B,C :: CALL SOU
ND(B*100,C,5,C+2,5):: NEXT A
830 FOR A=5 TO 30 :: CALL SO
UND(-99,196,A,198,A):: NEXT
A :: CALL T(999):: CALL DELS
PRITE(ALL):: GOTO 680
840 CALL LOCATE(#1,X1,Y1)::
IF JA<125 THEN CALL SOUND(99
9,-3,0):: CALL T(999):: GOTO
 880
850 DATA 4,4,3,3,13,13,3,3
860 FOR A=1 TO 3 :: CALL SOU
ND(-4250,-8,0):: RESTORE 850
 :: FOR B=1 TO 8 :: READ C :
: CALL COLOR(#1,C):: NEXT B
:: CALL SOUND(-1,-5,19)
870 NEXT A :: CALL COLOR(#1,
4):: CALL SAY("E+T LIKES+THA
T"):: JA=0 :: G=G+150 :: CAL
L K(G)
880 CALL LOCATE(#1,X1,Y1-8):
: GOTO 240
890 CALL LOCATE(#1,X1,Y1)::
IF KE=1 THEN CALL SAY("IT+IS
 O+K I+HAVE+IT"):: CALL F(98
2):: CALL F(880):: CALL F(83
1):: CALL F(784):: CALL F(74
0):: GOTO 250
900 FOR A=0 TO 30 STEP 2.5 :
: CALL SOUND(-99,220,A,330,A
):: NEXT A :: FOR A=0 TO 30
STEP 5 :: CALL SOUND(-99,220
,A,196,A):: NEXT A
910 FOR A=0 TO 30 STEP 5 ::
CALL SOUND(-99,196,A,330,A):
: NEXT A :: FOR A=1 TO 2 ::
FOR B=0 TO 30 STEP 2.5 :: CA
LL SOUND(-99,330,B,220,B)::
NEXT B
920 NEXT A :: CALL LOCATE(#1
,X1,Y1+8):: CALL SAY("E+T HA
VE+KEY"):: KE=1 :: G=G+200 :
: CALL K(G):: GOTO 250
930 CALL LOCATE(#1,X1,Y1)::
IF BE<250 THEN CALL SOUND(99
9,-2,0):: CALL T(999):: GOTO
 970
940 DATA 4,3,13,14,7,9,10,11
,12,15,16,15,12,11,10,9,7,14
,13,3,4
950 FOR A=1 TO 2 :: RESTORE
940 :: FOR B=1 TO 21 :: READ
 C :: D=INT(RND*9)+3 :: CALL
 SOUND(-333,-4,D):: CALL COL
OR(#1,C):: NEXT B :: NEXT A
960 CALL SAY("X"):: BE=0 ::
G=G+150 :: CALL K(G)
970 CALL LOCATE(#1,X1,Y1+8):
: GOTO 250
```

LISTINGS

```
980 SUB Z :: CALL HCHAR(1,1,
32,768):: CALL HCHAR(24,1,68
,32):: CALL VCHAR(3,1,69,22)
:: CALL VCHAR(3,32,69,22)
990 DISPLAY AT(3,1):"DDDHDDH
DDDDHDDDDDHDDDDHDDHDD...E..E
....E.....E....E..E.....E..E
....E.....E....E..E..JQQLQQL
QQKDIDDJQQBDJQQLQQLQQ"
1000 DISPLAY AT(9,1):"QQKA..
@DDDDDAQQ@DDDDDA..@JQQ...E..
E.....L..L.....E..E......E..
E.....P..P.....E..E......E..
E.....P..P.....E..E"
1010 DISPLAY AT(13,1):"...E.
.GDDDDDJ..KDDDDDF..E...QQKF.
.E.....P..P.....E..GJQQ...L.
.E.....P..P.....E..L......P.
.E.....M..M.....E..P"
1020 DISPLAY AT(17,1):"...P.
.EQQ@DDIDDIDDAQQE..P......M.
.L..E........E..L..M...QQKF.
.P..E........E..P..GJQQ...E.
.P..E........E..P..E......E.
.KDDC........BDDJ..E"
1030 DATA 3,1,64,3,2,68,3,31
,68,3,32,65,6,1,71,6,2,68,6,
31,75,6,32,70,9,1,71,9,2,74,
9,31,75
1040 DATA 9,32,70,14,1,71,14
,2,74,14,31,75,14,32,70,19,1
,71,19,2,74,19,31,75,19,32,7
0,24,1,66
1050 DATA 24,6,73,24,27,73,2
4,32,67,22,6,69,23,6,69,22,2
7,69,23,27,69
1060 DATA 19,16,46,19,17,47,
20,16,125,20,17,127,21,15,12
4,21,16,126,21,17,44,21,18,4
5
1070 DATA 22,15,116,22,16,11
8,22,17,120,22,18,122,23,15,
117,23,16,119,23,17,121,23,1
8,123
1080 RESTORE 1030 :: FOR A=1
 TO 44 :: READ B,C,D :: CALL
 HCHAR(B,C,D):: NEXT A :: A=
0
1090 CALL KEY(0,K,S):: IF S=
0 THEN 1100 ELSE 1110
1100 A=A+1 :: IF A>=300 THEN
 1110 ELSE 1090
1110 FOR A=15 TO 18 :: CALL
VCHAR(19,A,32,5):: NEXT A ::
 CALL F(456):: SUBEND
1120 SUB H(PAT):: Z=INT(RND*
2)+1 :: ON Z GOTO 1130,1140
1130 PAT=88 :: CALL MOTION(#
6,0,-4):: SUBEXIT
1140 PAT=96 :: CALL MOTION(#
6,0,4):: SUBEND
```

```
1150 SUB P
1160 DATA 5,2,116,4,31,45,13
,2,121,11,11,124,11,23,47,12
,30,125,17,4,126,20,11,123,2
0,22,119,17,29,46
1170 DATA 15,22,44,23,4,122,
23,30,120,4,7,118,4,27,127,1
6,12,117
1180 RESTORE 1160 :: FOR A=1
 TO 16 :: READ B,C,D :: CALL
 HCHAR(B,C,D):: NEXT A :: SU
BEND
1190 SUB K(G):: G$=STR$(G)::
 DISPLAY AT(1,3)SIZE(LEN(G$)
):G$ :: SUBEND
1200 SUB Q
1210 CALL CHAR(88,"00031E704
05F404FC09EA1C0A19EC0407EC50
2020CF000008E51149600324181"
,92,"FC478080601F0001E21441D
211788402008OF01C04F404E406F
20A060AF20604")
1220 CALL CHAR(96,"414161203
E0388D8A8888A202020203A01403
28D40C0705F4028271012090804"
,100,"0105996204071CF40412E2
0105F800020404CC08F880000000
00000000808040")
1230 CALL CHAR(38,"0000040E0
4",39,"000040E04",60,"3C4299
A1A199423C")
1240 CALL CHAR(104,"41416120
3E0388D8A8888A202020203A4032
2D904CC3705F402024131OOA0904
",108,"0599691264871CF404020
AF10104F80004040C08F88000000
000000000808040")
1250 CALL COLOR(0,5,5,1,5,12
,2,7,12,3,7,12,4,7,12,5,7,12
,6,7,12,7,7,12,8,7,12):: CAL
L SCREEN(5):: FOR A=1 TO 24
:: PRINT :: NEXT A
1260 CALL VCHAR(1,3,30,24)::
 CALL D(2,3,"MANFRED LIPOWSK
I PRESENTS"):: CALL D(4,14,"
E.T."):: CALL D(6,4,"THE E=T
RA-TERRESTIAL IN")
1270 CALL D(8,4,"HIS ADVENTU
RE ON EARTH"):: CALL D(10,7,
"< SEPTEMBER 1985"):: CALL F
(208):: CALL F(311):: CALL F
(208):: CALL HCHAR(15,16,38)
1280 CALL HCHAR(15,19,39)::
CALL SPRITE(#1,88,2,105,121,
#2,92,2,105,137,#3,96,2,121,
121,#4,100,2,121,137)
1290 CALL D(22,5,"PRESS AN>
KE> TO BEGIN")
1300 DATA 5,6,7,9,10,8,8,10,
9,7,6,5
1310 D=INT(RND*9)+1 :: FOR C
=1 TO D :: RESTORE 1300 :: F
OR A=1 TO 12 :: READ B :: CA
LL COLOR(1,B,12):: CALL KEY(
0,K,S):: IF S<>0 THEN 1350
1320 NEXT A :: NEXT C :: CAL
L PATTERN(#3,104,#4,108):: C
ALL SAY("E"):: CALL PATTERN(
#3,96,#4,100):: CALL T(25)
1330 CALL PATTERN(#3,104,#4,
108):: CALL SAY("T"):: CALL
PATTERN(#3,96,#4,100):: CALL
 T(25):: CALL PATTERN(#3,104
,#4,108):: CALL SAY("WILL")
1340 CALL PATTERN(#3,96,#4,1
00):: CALL T(25):: CALL PATT
ERN(#3,104,#4,108):: CALL SA
Y("HOME"):: CALL PATTERN(#3,
96,#4,100):: GOTO 1310
1350 CALL COLOR(1,5,12):: SU
BEND
1360 SUB W :: CALL CHAR(44,"
3CE4E6FF03FF0FFE0A040464F498
F87050211D8743463E030A84B8E1
C2627CC0")
1370 DATA 0000000000002070F8
3F1F1F0C142244000000000000000
0204F8FC2C2C444284,000000000
00010387C1F0F1F0C040A1100000
0000000000804F8FC1C0C0A1109
1380 DATA 0000000000000004020
1F3F34342242210000000000000004
0E1FFCF8F830284422,000000000
0000010201F3F383050889000000
0000000081C3EF8F0F830205088
1390 DATA 0705070300010307OB
1317170703020FF0D0F0E080C0E0
F0E8E474F4F0E02078,030203010
001030F130B07070703020FF8E8F
8F080C0E0F0E8E4B2F1F0E01418
1400 DATA 1F171F0F0103070F17
274D8F0F072818C040C0800080C0
F0C8D0E0E0E0C040F,4781516143
464D7B3B0D060301192573FEE3FF
9F9CFBFAFAABFCDFDCF8F90BFF
1410 DATA 7FC7EDE139DF5B5FDF
3BFB3F159FD0FFE2818A86C262B2
DEDCBC60C08098A4CE,502020262
F191F0E0F1F3C6B3A1A1A3B3C276
7FFC0FFF07FF0F83CD65C5858DC
1420 DATA 7EDFFF7F0301030F53
27050703010107000000808OC0E0
F0E8E4E2C2C0A810B,3F6F7F3F00
010303031 70E070302140D008080
C0C0C0E0F0E890E0E0C0C040E
1430 DATA 0000000101030 70F17
2747430315080D7EFBFFFEC080C0
F0CAE4A0E0C08080E,0001010303
03070F17090707030302 07FCF6FE
FC0080C0C0C0E870E0C04028B
1440 RESTORE 1370 :: FOR A=8
8 TO 140 STEP 4 :: READ A$ :
```

```
: CALL CHAR(A,A$):: NEXT A
1450 CALL CHAR(64,"FFFFC0C0C
0C0C3C3FFFF0303030303C3C3C3C3C
0C0C0C0FFFFC3C303030303FFFF"
,68,"FFFF00000000FFFFC3C3C3C
3C3C3C3C3C3C303030303C3C3C3C
3C0C0C0C0C3C3")
1460 CALL CHAR(60,"1C22221C0
8080808080818083808",40,"7EF
FFF9BAF9BAB9BFF93B593B593FF7
E",36,"001F2040FF8AEAEAAADDF
FCAEAAADDFF00FF0305F9A989A9A
9A9F9A989AAACF8")
1470 CALL CHAR(72,"FFFF00000
000C3C3C3C300000000FFFF",80,
"1818181818181818000000FFFF"
,74,"FFFF03030303FFFFFFFFC0C
0C0C0FFFF")
1480 CALL CHAR(76,"C3C3C3C3C
3C3FFFFFFFFC3C3C3C3C3C3")
1490 CALL CHAR(62,"",63,"",4
2,"",43,"",82,""):: SUBEND
1500 SUB T(Z):: FOR Z1=1 TO
Z :: NEXT Z1 :: SUBEND
1510 SUB D(Z,Z1,A$):: DISPLA
Y AT(Z,Z1):A$ :: SUBEND
1520 SUB F(TON):: FOR A=0 TO
 30 STEP 5 :: CALL SOUND(-99
,TON,A,TON+2,A,TON-2,A):: NE
XT A :: SUBEND
```

LISTINGS

RETTET DIE ROBBEN!

Dieses Action-Spiel ist eine Mischung aus Frogger, Donkey Kong und einem neuartigen Spiel.
Ziel ist es, mit Hilfe eines kleinen Männchens Robbenbabies vor den Jägern zu retten. Dazu müssen Sie im ersten Screen zum Haus der Robbenjäger übers Eismeer gelangen, im zweiten Screen die im Haus versteckte Farbe zum Besprühen der Tiere finden und im dritten Screen die Babies einfärben. Daß dies nicht ohne Hindernisse, Schwierigkeiten und Gefahren möglich ist, versteht sich wohl von selbst. Bei jeder mißlungenen Action verlieren Sie lebenswichtige Energie. Der Energievorrat wird in Säulenform am rechten Bildschirmrand angezeigt. Wenn Sie einen Durchgang erfolgreich beendet haben, bekommen Sie von der Robbenkolonie einen Orden, der Energievorrat erhöht sich um einen bestimmten Betrag, und die nächste Runde beginnt.
Machen Sie den Robbenkillern das Leben schwer!

BÖRSE

Verschiedene Module mit Spielen und Anwendungen: Aktuell und billig: Assemblerpaket mit Handbuch (dtsch.), Demo-Kassette und natürlich mit Steckmodul zum direkten Anschluß an die Konsole + Ext. basic für nur DM 189,00. Externe Schnittstelle RS 232 (V24EA) mit sehr gutem Beschrieb und Kabel alles für DM 170. Drukker, EP 22, Fabr. Brother, mit Interface, Display, auch als Schreibmaschine zu nutzen, nur DM 320. Bücher mit Tips + Tricks, je DM 30. Kassetten mit div. Programmen je DM 15. Peter Kruber, Schloßstraße 72, 6908 Wiesloch, Telefon 06222/53407.

TI99/4A + Data-Rec. + Bas.-Lehrg. + 100 Progr. + Data-Becker Buch 345 DM * Wx + .-Bas.-Modul + Lehrg. + 50 Ext. Progr. 225 DM * 16 Module (Schach, Musik-Maker usw.) sowie TI-Cassetten + Bücher sowie 16 Comp.-Zeitschriften *. Telefon 02174/40654.

TI 99/4A + Pe. Box + Floppy Cont. + Floppy Laufw. + 32 K Ram + RS 232 + Speech Synt. + X-Basic + Ed/As. + Schachm. + Joyst. + Rec. Kabel + Recorder + viel Literatur + viel Maschinen- + Basic Progs. (z. B. Ti Forth, 3D-World) DM 2.000, bei Rainer S. Telefon 02225/13581.

Suche billigen, aber heilen Datenrecorder mit Kabel bis DM 40. Marco Neumann, Jungborn 42, 2000 Hbg. 6

TI-99/4 A-Super-Spiele auf Kas. M. Boron: Hochenstaufenstr. 30, 7141 Möglingen, Tel.: 07141/461843. PS: Viel mehr als Spiele.

Verkaufe: 99/A mit Exb, MM, Invard., Joyst., CS1 & CS2, div. Pgme, XB-, Ass, 99 Spez. I&II, MM-Handbücher, alle deutsch! Sowie div. Zeitschriften. VB 900 DM, mögl. kompl, evtl. einzeln. Tel.: 040/677 27 39 ab 19.00

Cad-Katalog 86 !!!!!!!!!!!!!! 22 Firmen, Adressen, Soft- und Hardware-Beschreibung von Cad-Systemen, Preise incl. Gratis Dateiverwaltung (Daten lesen, schreiben, suchen). Kas.: 60,-- DM. K. Gaisser, Fuchsfeldweg 9, 7155 Zell, Tel.: 07191/44211

Verk.: TI 99/4A + Box + 32 K + LW + Contr + Ex-Basic+ Rec-Kabel + Lit. + Joyst., VB 1400,-- DM. Tel.: 0221/5906379,

---

Verk. wegen Hobbywechsel sehr günstig mein System! Gratisliste: B. Leoni, Rainstr. 19d, CH - 8808 Pfäffikon

---

Verkaufe oder tausche meine 32 KRAM mit Centronics (Extern) gegen origin. 32 KRAM-Karte. Tel.: 02732/704325 – A.

Datenverw. u. Analyse-Modul f. TI 99/4A zu verkaufen 100,-- DM . S. Bauer, Tannenweg 18, 7180 Crailsheim.

Verkaufe: Personal Record Keeping Statistics. German Pers. Report Generator. Preis-VB. Tel.: 08431/45808

Tausche Korg D DM 110, Digital-Drums gegen Peribox + LW + Contr. OD Ext. LW + Contr., NP 90 DM, 4 Monate alt. M. Lehn, Hubertusring 29, 3180, Wolfsburg 23

Verkaufe folgende TI-Module Text & Dateiverwaltung 30,-- DM, Buchungsjournal 50,-- DM, Statistik 30,-- DM, Datenverwaltung & Anal. 35,-- DM, Schachmeister 40,-- DM, Adventure-Modul 40,-- DM, Music Maker 40,-- DM, Othello 25,-- DM, Editor Assembler 110,-- DM. BASF Laufwerk 6106 200,-- DM, Org. TI-Laufwerk 200,-- DM. Tel.: 0212/56537 oder 43140.

Tausche: Disk. Controler + Modul gegen Ex-Basic Modul Th. Frick, Spitalgasse 4, 8001 Uerich/Schweiz.

LISTINGS

```
10 ! **********************
11 ! *                    *
12 ! *    ROBBENRETTER    *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *    Peter Avgath    *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *     Joystick 1     *
23 ! *    Cassettenrec.   *
24 ! *(oder Disk+32k-Erw.)*
25 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *    12034 Bytes     *
28 ! *                    *
29 ! **********************
100 CALL CHAR(92,"000103060C
183060C080FF7F3F1F0F07C7837F
414121110907 01FFFFFFFFFFFF")
110 CALL CHAR(96,"E0C0FE8282
848890E080FFC7DFD7C5FF07037F
414121110907 01FF8FAF8FBBFF")
120 CALL CHAR(100,"E0C0FE828
2848890E080FFFFFFFFFFFF80804
0402020100804 02FFFFEFCF8F8"
)
130 CALL CHAR(104,RPT$("0",4
8)&"0103070D193163C7")
140 CALL CHAR(108,RPT$("0",1
6)&"80C0BF90909FC0E0"&RPT$("
0",16)&"0103FD0909F90307")
150 CALL CHAR(112,RPT$("0",1
6)&"00FC84828281C1E0"&RPT$("
0",30)&"80")
160 CALL CHAR(116,"000103050
F1F0301070911010202020600 80C
0A0F0F8C080E0908880404040C0"
)
170 CALL CHAR(120,"000000000
001023C4281A5818199423C00000
00000F8040202010101020 78A72"
)
180 CALL CHAR(128,"2121914A2
0070F1C1C1C1C1C1C0F070012222
44900E0F0F8F8F8F83838F0E000"
)
190 CALL CHAR(132,"00000F3F3
F78FCFC84BCA4B484FCFCFC40C0A
0A0A0C0400000")
200 CALL CHAR(136,"000000009
048243F3F2448900000000000000
000000406FFFF0604")
210 CALL CHAR(140,"000000000
02060FFFF6020000000000000000
000091224FCFC24120900")
220 RANDOMIZE :: CALL MAGNIF
Y(2):: CALL CLEAR :: CALL SC
REEN(2):: FOR I=0 TO 14 :: C
ALL COLOR(I,16,1):: NEXT I
230 CALL COLOR(10,5,1,11,5,1
)
240 A$="RROEBTBTEENR"
250 FOR I=1 TO 12 STEP 4 ::
FOR II=1 TO 4 :: CALL CHARPA
T(ASC(SEG$(A$,(I-1)+II,1)),B
$):: C$(I/4)=C$(I/4)&B$ :: N
EXT II :: NEXT I
260 CALL CHAR(52,C$(0),56,C$
(1),60,C$(2))
270 CALL MAGNIFY(4):: CALL S
PRITE(#13,120,11,98,100,#14,
136,6,100,1,0,4,#15,140,6,10
0,199,0,-4)
280 DISPLAY AT(1,1):"SOLO-SO
FT" :: CALL SOUND(-200,261,0
,259,4,263,8):: FOR I=1 TO 1
00 :: NEXT I
290 DISPLAY AT(2,1):"PROUDLY
" :: CALL SOUND(-200,329,0,3
27,4,331,8):: FOR I=1 TO 100
 :: NEXT I
300 DISPLAY AT(3,1):"PRESENT
S!" :: CALL SOUND(-200,391,0
,389,4,393,8):: FOR I=1 TO 1
00 :: NEXT I
310 CALL SOUND(-1000,523,0,5
21,4,525,8)
320 CALL SPRITE(#1,52,6,50,4
0,#2,56,6,50,72,#3,60,6,50,1
04)
330 DISPLAY AT(24,1):"ES WAR
 EINMAL..."
340 CALL COINC(ALL,C):: IF C
 THEN 350 ELSE 340
350 CALL DELSPRITE(ALL):: FO
R I=0 TO 30 STEP 5 :: CALL S
OUND(-1,110,20,112,20,4000,3
0,-8,I):: NEXT I
360 DISPLAY AT(1,1)ERASE ALL
:"... EIN KLEINES MAENNLEIN,
": :"DAS FUHR MIT SEINEM": :
"SEGELSCHIFF ZUM NORDPOL,":
:"UM DIE ROBBENBABIES"
370 DISPLAY AT(9,1):"VOR DEN
 ROBBENJAEGERN ZU": :"SCHUET
ZEN.ES MUSS DAZU": :"VON SEI
NEM SCHIFF UEBER DIE": :"EIS
BERGE AN LAND HUEPFEN."
380 DISPLAY AT(17,1):"ES IST
 IHRE AUFGABE,IHM": :"DABEI
ZU HELFEN.": :"SOBALD ER AN
LAND IST,": :"KOMMEN WEITERE
 ANWEISUNGEN."
390 CALL KEY(1,K,S):: IF S=0
 THEN 390 ELSE CALL CLEAR
400 !
410 ! ROBBENSPIEL TEIL 1
420 !
430 CALL MAGNIFY(3):: STUFE=
```

LISTINGS

```
1 :: EN=190 :: RANDOMIZE ::
CALL ASCII(EN)
440 FOR I=8 TO 11 :: CALL CO
LOR(I,13,8):: NEXT I
450 II=2 :: FOR I=2 TO 11 ST
EP 3 :: II=II+1 :: B=4*(8-II
):: IF II=3 OR II=5 THEN B=-
B
460 CALL SPRITE(#I,124,1,II*
24,80,0,B,#I+1,124,1,II*24,1
60,0,B,#I+2,124,1,II*24,240,
0,B):: NEXT I
470 CALL SPRITE(#14,92,13,17
6,208,#15,96,13,176,222,#16,
100,13,176,236)
480 CALL SPRITE(#17,104,13,1
60,206,#18,108,13,160,220,#1
9,112,13,160,234)
490 FOR I=14 TO 19 STEP 2 ::
 CALL MOTION(#I,0,-8,#I+1,0,
-8):: NEXT I
500 FOR I=1 TO 400 :: NEXT I
 :: CALL SOUND(-300,391,0,38
9,4,393,0)
510 CALL DELSPRITE(#14,#15,#
16,#17,#18,#19)
520 DISPLAY AT(21,12)SIZE(6)
:"hjlnpr" :: DISPLAY AT(22,1
2)SIZE(6):"ikmoqs"
530 DISPLAY AT(23,12)SIZE(6)
:"\^`bdf" :: DISPLAY AT(24,1
2)SIZE(6):"]_aceg"
540 CALL SPRITE(#28,108,13,1
61,121,#1,116,13,160,122)
550 FOR I=2 TO 13 :: CALL CO
LOR(#I,16):: NEXT I
560 CALL COINC(ALL,C):: IF C
=0 THEN 640
570 CALL KEY(1,K,S):: IF S=0
 THEN 570
580 CALL JOYST(1,X3,Y3):: IF
 X3=0 AND Y3=0 THEN 570
590 CALL POSITION(#1,X2,Y2):
: IF X2<80 THEN 660
600 X=Y3/4 :: Y=X3/4 :: FOR
I=1 TO 23 STEP 11 :: CALL LO
CATE(#1,X2-I*X,Y2+I*Y):: NEX
T I
610 CALL POSITION(#1,X2,Y2):
: MO=INT(6-((X2-50)/24)):: I
F MO/2<>INT(MO/2)THEN MO=-MO
620 CALL MOTION(#1,0,4*MO)
630 GOTO 560
640 CALL ENERGIE(EN):: CALL
SOUND(100,-7,0):: CALL SPRIT
E(#1,116,13,160,122,0,0):: G
OTO 560
650 CALL DELSPRITE(ALL)
660 CALL LOCATE(#1,40,Y2)::
CALL MOTION(#1,0,30):: FOR I
=1 TO 310-Y2 :: NEXT I :: CA
LL MOTION(#1,0,0)
670 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL CHARSET ::
CALL SCREEN(2):: FOR I=0 TO
14 :: CALL COLOR(I,11,1):: N
EXT I
680 CALL CHAR(40,"FFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFC0C0C0FFFFC0C0C0")
690 CALL CHAR(45,"030303FFFF
030303",59,"EEDDBB77EEDDBB77
",47,"00EEEEEEEEEEEEEE")
700 DISPLAY AT(1,1)ERASE ALL
:"ES FOLGT TEIL 2": :"((((((
((((((((((((((((((((("
710 DISPLAY AT(5,1):"ER DRIN
GT IN DAS HAUS": :"DER ROBBE
NJAEGER EIN,": :"UND SUCHT D
AS FARBSPRAY.":"DOCH DA GIBT
 ES NOCH":"LEBERTRANTONNEN!"
720 DISPLAY AT(12,10):")*" :
: DISPLAY AT(13,9):")((*" ::
 DISPLAY AT(14,8):")((((*"
730 DISPLAY AT(15,8):"+-+-+-
" :: DISPLAY AT(16,8):"+-+-+
-"
740 CALL SPRITE(#1,116,13,1,
1,40,20)
750 FOR I=1 TO 160 :: NEXT I
 :: CALL MOTION(#1,0,30):: F
OR I=1 TO 45 :: NEXT I :: CA
LL MOTION(#1,-20,0):: FOR I=
1 TO 30 :: NEXT I
760 CALL MOTION(#1,0,0):: CA
LL KEY(1,K,S):: IF S=0 THEN
760
770 !
780 !ROBBENSPIEL TEIL 2
790 !
800 X=1 :: Y=10 :: STUFE=STU
FE+.2
810 CALL CLEAR :: CALL COLOR
(4,7,2):: CALL LOCATE(#1,X,Y
)
820 NR=NR+1 :: IF NR>2 THEN
NR=1
830 CALL HAUS(NR,E(),EN)
840 D=((NR=2)*2+1)*STUFE
850 CALL SPRITE(#3,128,8,55,
80,0,-8*D,#4,128,8,55,160,0,
-8*D,#5,128,8,55,240,0,-8*D)
860 CALL SPRITE(#6,128,6,110
,60,0,12*D,#7,128,6,110,140,
0,12*D,#8,128,6,110,220,0,12
*D)
870 CALL SPRITE(#9,128,5,166
,40,0,-12*D,#10,128,5,166,12
0,0,-12*D,#11,128,5,166,200,
0,-12*D)
880 CALL SPRITE(#2,132,4,130
,240)
```

LISTINGS

```
890 CALL JOYST(1,X2,Y2)
900 CALL MOTION(#1,0,2*X2)
910 CALL LEITER(X,Y,E())
920 CALL KEY(1,K,S):: IF S=0
 THEN CALL COINCALL(EN):: GO
TO 890
930 CALL MOTION(#1,0,0):: CA
LL JOYST(1,X2,Y2):: CALL POS
ITION(#1,X,Y)
940 IF X-20<1 OR Y+8*X2<1 OR
 Y+8*X2>256 THEN 890
950 FOR I=1 TO 30 STEP 10 ::
 CALL LOCATE(#1,X-I,Y+(I/6)*
X2):: A=EN :: CALL COINCALL(
EN)
960 IF A<>EN THEN CALL LOCAT
E(#1,X,Y+(I/6)):: GOTO 890
970 NEXT I
980 CALL COINC(#1,#2,20,C)::
 IF C THEN CALL DELSPRITE(AL
L):: GOTO 1010
990 I2=5 :: FOR I=30 TO 0 ST
EP -10 :: I2=I2+.5 :: CALL L
OCATE(#1,X-I,Y+I2*X2):: CALL
 COINCALL(EN):: NEXT I
1000 GOTO 890
1010 CALL CLEAR :: FOR I=0 T
O 14 :: CALL COLOR(I,16,1)::
 NEXT I
1020 DISPLAY AT(1,1):"NUN FO
LGT DER DRITTE TEIL":"((((((
((((((((((((((((((((((((": :"E
R MUSS DIE ROBBENBABIES": :"
MIT DER FARBE BESPRUEHEN,"
1030 DISPLAY AT(8,1):"INDEM
ER SIE BERUEHRT.": :"DIE ROB
BENJAEGER BESCHIESSEN": :"IH
N WAEHRENDDESSEN MIT ": :"PF
EILEN,DENEN ER NATUERLICH"
1040 DISPLAY AT(16,1):"AUSZU
WEICHEN HAT."
1050 CALL KEY(1,K,S):: IF S=
0 THEN 1050 ELSE 1090
1060 !
1070 ! ROBBENSPIEL TEIL 3
1080 !
1090 CALL DELSPRITE(#1):: CA
LL CLEAR :: ROBBE=6 :: S=STU
FE
1100 CALL ASCII(EN):: CALL C
OLOR(1,16,1,12,2,16,13,2,16)
:: EN=EN+5 :: CALL ENERGIE(E
N)
1110 FOR I=2 TO 10 STEP 4 ::
 CALL SPRITE(#I,140,5,I*10+4
5,80,0,-10*S,#I+12,140,5,I*1
0+45,240,0,-10*S)
1120 CALL SPRITE(#I+17,120,1
4,I*10+45,120,0,-10*S)
1130 CALL SPRITE(#I+2,136,5,
(I+2)*10+45,60,0,10*S,#I+11,
136,5,(I+2)*10+45,220,0,10*S
)
1140 CALL SPRITE(#I+14,120,1
4,(I+2)*10+45,100,0,10*S)::
NEXT I
1150 CALL SPRITE(#1,116,13,1
80,128)
1160 CALL JOYST(1,X2,Y2):: C
ALL MOTION(#1,-Y2*3,X2*3)::
CALL COINC(ALL,C):: IF C=-1
THEN 1180
1170 GOTO 1160
1180 CALL MOTION(#1,0,0):: F
OR I=16 TO 24 STEP 4 :: CALL
 COINC(#1,#I,16,C):: CALL CO
INC(#1,#I+3,16,C2):: IF C=-1
 OR C2=-1 THEN 1200
1190 NEXT I :: CALL SOUND(50
,-6,0):: CALL ENERGIE(EN)::
GOTO 1160
1200 IF C=-1 THEN CALL POSIT
ION(#I,XX,YY):: CALL DELSPRI
TE(#I)ELSE CALL POSITION(#I+
3,XX,YY):: CALL DELSPRITE(#I
+3)
1210 ROBBE=ROBBE-1 :: EN=EN+
20 :: IF EN>190 THEN EN=190
1220 CALL ENERGIE(EN):: IF R
OBBE=0 THEN CALL DELSPRITE(A
LL):: GOTO 1280
1230 CALL SOUND(100,329,0,32
7,4,331,8):: CALL SOUND(300,
393,0,391,4,395,8)
1240 X1=INT(XX/8)+1 :: Y1=IN
T(YY/8)+1 :: IF X1>23 THEN X
1=23
1250 IF Y1>31 THEN Y1=23
1260 CALL HCHAR(X1,Y1,120)::
 CALL HCHAR(X1,Y1+1,122):: C
ALL HCHAR(X1+1,Y1,121):: CAL
L HCHAR(X1+1,Y1+1,123)
1270 GOTO 1160
1280 CALL VIC(EN):: GOTO 670
1290 SUB ENERGIE(EN):: CALL
COLOR(4,7,2)
1300 EN=EN-10 :: IF EN>8 AND
 EN<=190 THEN H1=INT(EN/8)::
 H2=EN-H1*8 ELSE IF EN<8 THE
N CALL NDLG ELSE SUBEXIT
1310 CALL CHAR(60,RPT$("F",1
6),62,"1010101C1010101C",63,
"")
1320 CALL VCHAR(1,31,62,24):
: CALL VCHAR(1,32,63,24-(H1+
1))
1330 CALL CHAR(61,RPT$("00",
8-H2)&RPT$("FF",H2))
1340 CALL VCHAR(25-H1,32,60,
H1)
1350 CALL HCHAR(24-H1,32,61)
1360 SUBEND
```

LISTINGS

```
1370 SUB COINCALL(EN)
1380 CALL COINC(ALL,C):: IF
C=-1 THEN CALL ENERGIE(EN)
1390 SUBEND
1400 SUB LEITER(X,Y,E())
1410 CALL POSITION(#1,X,Y)
1420 IF (X=1 AND(Y>E(1)AND Y
<E(2)))OR(X=56 AND(Y>E(3)AND
 Y<E(4)))OR(X=111 AND(Y>E(5)
AND Y<E(6)))THEN 1440
1430 GOTO 1460
1440 CALL JOYST(1,X2,Y2):: I
F X2=0 AND Y2=-4 THEN 1450 E
LSE 1460
1450 FOR I=1 TO 55 STEP 6 ::
 CALL LOCATE(#1,X+I,Y):: CAL
L SOUND(-10,-6,0):: NEXT I :
: X=X+40
1460 SUBEND
1470 SUB HAUS(NR,E(),EN)
1480 FOR I=2 TO 23 STEP 7 ::
 CALL HCHAR(I,1,59,30):: CAL
L HCHAR(I+1,1,47,30):: NEXT
I
1490 IF INT(NR/2)<>NR/2 THEN
 1500 ELSE 1550
1500 CALL VCHAR(3,5,43,7)::
CALL VCHAR(3,6,45,7):: CALL
VCHAR(10,25,43,7):: CALL VCH
AR(10,26,45,7)
1510 CALL VCHAR(3,5,43,7)::
CALL VCHAR(3,6,45,7):: CALL
VCHAR(10,25,43,7):: CALL VCH
AR(10,26,45,7)
1520 CALL VCHAR(17,5,43,7)::
 CALL VCHAR(17,6,45,7)
1530 E(1)=30 :: E(2)=40 :: E
(3)=190 :: E(4)=200 :: E(5)=
30 :: E(6)=40
1540 GOTO 1590
1550 CALL VCHAR(3,10,43,7)::
 CALL VCHAR(3,11,45,7):: CAL
L VCHAR(10,4,43,7):: CALL VC
HAR(10,5,45,7)
1560 CALL VCHAR(3,10,43,7)::
 CALL VCHAR(3,11,45,7):: CAL
L VCHAR(10,4,43,7):: CALL VC
HAR(10,5,45,7)
1570 CALL VCHAR(17,25,43,7):
: CALL VCHAR(17,26,45,7)
1580 E(1)=70 :: E(2)=80 :: E
(3)=20 :: E(4)=30 :: E(5)=19
0 :: E(6)=200
1590 EN=EN+5 :: CALL ENERGIE
(EN):: SUBEND
1600 SUB ASCII(EN)
1610 Q=Q+1 :: IF Q>1 THEN CA
LL COLOR(1,16,1,12,2,16)ELSE
 CALL COLOR(1,8,1)
1620 FOR I=2 TO 14 :: CALL C
OLOR(I,15,1):: NEXT I
1630 CALL CHAR(33,RPT$("F",1
6))
1640 CALL CHAR(64,"000001070
F1F7FFF7FFFFFFFFFFFFFFF00000
00001070F1F1F7FFFFFFFFFFFFF"
)
1650 CALL CHAR(68,"000000000
3070F1F000000F0F8F8FCFF80C0F
0F8FCFDFFFF0000000080C0E0F0"
)
1660 CALL CHAR(72,"FCFEFFFFF
FFFFFFF0E7FFFFFFFFFFFFF0183C
7EFFFFFFFFF0103070F0F1F7FFF"
)
1670 CALL CHAR(76,"070F0F1F3
F7F7FFF000000001C3C7EFF80C0F
8FFFFFFFFFF00000000C0F0FEFF"
)
1680 CALL CHAR(80,"81C3E7EFF
FFFFFFF000000103C7CFEFFFFFFF
FFFFFFFFFFF")
1690 CALL CHAR(124,"0303070D
1B1B3F2E2F5F777BFDFDFFFF80C0
60A0B0F8F8F87C76F6FAFFFFFFF
")
1700 DISPLAY AT(3,1):".....
...BG"
1710 DISPLAY AT(4,1):"....M.
..LRHG.....BCF........."
1720 DISPLAY AT(5,1):"...KRF
 KRRRHIJNOBCRRFGKNO..."
1730 DISPLAY AT(6,1):"..KRRR
PRRRRRRRRRRRRRRRRF.."
1740 DISPLAY AT(7,1):"DIRRRR
RRRRRRRRRRRRRRRRRNO"
1750 FOR I=8 TO 24 :: CALL H
CHAR(I,1,33,30):: NEXT I ::
EN=EN+5 :: CALL ENERGIE(EN)
1760 SUBEND
1770 SUB VIC(EN)
1780 CALL CLEAR :: CALL CHAR
SET :: CALL SCREEN(2):: FOR
I=0 TO 14 :: CALL COLOR(I,11
,1):: NEXT I
1790 EN=EN+10 :: CALL ENERGI
E(EN)
1800 CALL SOUND(-500,261,0,2
59,4,263,8):: FOR I=1 TO 50
:: NEXT I
1810 H=H+1 :: H$=STR$(H)
1820 CALL SOUND(-500,329,0,3
27,4,331,8):: FOR I=1 TO 50
:: NEXT I
1830 DISPLAY AT(5,5):"B R A
V O..! ! !": : :"SIE HABEN E
S WIRKLICH": :"GESCHAFFT !!!
"
1840 CALL SOUND(-500,391,0,3
89,4,393,8):: FOR I=1 TO 50
:: NEXT I
1850 DISPLAY AT(12,1):"HIERM
```

```
IT WERDEN SIE ZUM": :"HELDEN
 DER ROBBEN "&H$&".KLASSE":
:"ERNANNT."
1860 CALL SOUND(-1000,523,0,
521,4,525,8):: FOR I=1 TO 25
0 :: NEXT I
1870 DISPLAY AT(18,1):"ABER
ES BLEIBT KEINE": :"ZEIT ZUM
 AUSRUHEN,": :"ES WARTEN NOC
H VIELE ROBBEN!"
1880 FOR I=1 TO 42 STEP 3 ::
 EN=EN+13 :: CALL ENERGIE(EN
):: CALL SOUND(-1000,110+I*5
,0,110+I*4,4,110+I*6,8):: NE
XT I
1890 EN=EN+50 :: IF EN>190 T
HEN EN=190
1900 CALL KEY(1,K,S):: IF S=
0 THEN 1900
1910 SUBEND
1920 SUB NDLG
1930 CALL DELSPRITE(ALL):: C
ALL CLEAR :: CALL CHARSET ::
 FOR I=0 TO 14 :: CALL COLOR
(I,14,1):: NEXT I :: CALL SC
REEN(2)
1940 DATA SCHWAECHLING,ALTE
FLASCHE,MIESE RATTE,NICHTSNU
TZ,TU-NICHT-GUT
1950 RESTORE 1940 :: A=INT(R
ND*5)+1 :: FOR I=1 TO A :: R
EAD A$ :: NEXT I
1960 DISPLAY AT(1,5):"DU ";A
$;"!!!": : :"DU HAST ES NICH
T GESCHAFFT,": :"DIE ROBBENB
ABIES ZU RETTEN.": :"DU HAST
 DIE WAHL ZWISCHEN"
1970 DISPLAY AT(10,1):"(1) 1
0 JAHREN ZUCHTHAUS": :"(2)..
5 JAHREN ZUCHTHAUS"
1980 CALL KEY(0,K,S):: IF K-
48<>1 AND K-48<>2 THEN 1980
1990 IF K-48=1 THEN DISPLAY
AT(14,1):"BRAVO,GUT GEWAEHLT
 !!!": :"SIE WERDEN WEGEN GU
TER": :"FUEHRUNG BEGNADIGT."
2000 IF K-48=1 THEN DISPLAY
AT(24,1):"NOCH EIN SPIEL J/J
 ?" :: ACCEPT AT(24,22):J$ :
: IF J$<>"J" AND J$<>"j" THE
N END ELSE RUN
2010 IF K-48=2 THEN DISPLAY
AT(14,1):"TUT MIR LEID FUER
SIE !!!": :"SIE BRINGEN EINE
N WAERTER": :"UM UND WERDEN
ZU LEBENS-"
2020 IF K-48=2 THEN DISPLAY
AT(20,1):"LAENGLICH VERURTEI
LT."
2030 CALL KEY(0,K,S):: IF S=
0 THEN 2030 ELSE END
2040 SUBEND
```

LISTINGS

TANK-COMMANDER

Das Programm Tankcommander wurde auf einem TI 99/4A in der Sprache Extended Basic geschrieben. Es werden 2 Joysticks und ein Sprach-Synthesizer benötigt.
Ein paar Sekunden nach dem Start erscheint ein Titelbild und der Computer fragt, ob eine Zeichenerklärung nötig ist. Wenn ja, dann braucht der Spieler nur noch die Taste "J" zu drücken.
Nachdem eine Taste gedrückt wurde, erscheint das Spielbild. In der drittletzten Zeile wird angezeigt, wieviele Panzer, Minenräumer und Schritte der Spieler 1 noch hat.
In der vorletzten Zeile wird ausgedruckt, wie viele Panzer, Minenräumer und Schritte der Spieler 2 noch hat. In der letzten Zeile, wie viel Zeit dem Spieler noch bleibt, bis ihn ein Flugzeug unter Beschuß nimmt.
Die blauen Fahrzeuge lenkt Spieler 1. Die schwarzen Fahrzeuge lenkt Spieler 2. Die blaue Partei muß die linke Zentrale vor dem Feind schützen. Die schwarze Partei muß die rechte Zentrale schützen. Das Ziel jedes Spielers ist es, die Zentrale zu zerstören. Wenn einer der Spieler keine Panzer und keine Minenräumer mehr hat, hat er verloren.

STEUERUNG

Jeder Spieler bedient einen Joystick. Wenn in der linken oberen Ecke ein rotes Visier erscheint, ist Spieler eins am Zug.
Wenn in der rechten oberen Ecke ein rotes Visier erscheint, ist Spieler zwei am Zug.
Wenn das Visier erscheint, kann je nachdem, welcher Spieler gerade dran ist, das Visier gelenkt und eines seiner Fahrzeuge anvisiert, sowie auf den Schußknopf gedrückt werden. Dann wird das Fahrzeug gelenkt. Mit der Taste "1" läßt man das Fahrzeug los. Mit der Taste "2" läßt sich der Turm des Panzers lenken. Auf den Minenwagen hat die Taste keinen Einfluß.
Wenn der Spieler einen Minenräumer lenkt und auf den roten Schußknopf drückt, löst sich hinter dem Minenwagen eine Mine. Sie ist für alle Panzer tödlich. Die Minenräumer hingegen können die Minen beseitigen.
Wenn der Spieler einen Panzer lenkt und auf den Schußknopf drückt, löst sich ein Schuß in die gezeigte Richtung.
Ich wünsche allen TI-Usern ein errorfreies Abtippen. Vor dem Laden des Programms eventuell CALL FILES (1) und NEW eingeben.

Funktionen:

180–400
Variablenbestimmung und Definitionen
420–800
Bilddruck
810–970
Joystickabfrage und Auswertung (Visier)
980–1450
Joystickabfrage und Auswertung (Panzer und Minenräumer)
1460–1660
Legen einer Mine oder Schuß und Auswertung
1670–1710
Errechnung und Ausdruck der Treffermeldung
1720–1810
Bewegung des Panzerturms Spieler 1
1820–1880
Bewegung des Panzerturms Spieler 2
1890–2120
Schuß des Panzerturms
2130–2260
Angriff eines Flugzeugs
2270–2400
Ende
2410–2780
Titelbild und Zeichenerklärung

Bitte lesen Sie weiter auf Seite 115

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *   TANK COMMANDER     *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Matthias Bosse     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *    Joystick 1&2      *
23 ! *   (Sprachsynth.)     *
24 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    11763 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CLEAR :: CALL DELSP
RITE(ALL):: ON BREAK NEXT
110 DIM PA(2),MI(2),SCH(2),Z
(2):: P=1 :: PA(1)=10 :: PA(
2)=10 :: MI(1)=4 :: MI(2)=4
:: SCH(1)=4 :: SCH(2)=4 :: Z
(1),Z(2)=15.00
120 CALL COLOR(1,11,11,2,3,1
1,9,5,11,11,2,11):: CALL SCR
EEN(2)
130 CALL CHAR(40,"9999666699
996666")
140 CALL CHAR(43,"100884FF84
081000")
150 CALL CHAR(118,"182442999
9422418")
160 CALL CHAR(41,"8124421818
422481")
170 CALL CHAR(42,"105055557D
7DFFFF")
180 CALL CHAR(47,"1010101010
101010")
190 CALL CHAR(96,"5500787F7F
780055")
200 CALL CHAR(112,"5500787F7
F780055")
210 CALL CHAR(97,"B530787878
780055")
220 CALL CHAR(98,"AA001EFEFE
1E00AA")
230 CALL CHAR(99,"AA00787878
7830B1")
240 CALL CHAR(113,"B53078787
8780055")
250 CALL CHAR(114,"AA001EFEF
E1E00AA")
260 CALL CHAR(115,"AA0078787
87830B1")
270 CALL CHAR(100,"8124183C3
C182481")
280 CALL CHAR(116,"8124183C3
C182481")
290 CALL CHAR(101,"AAAAFF818
1FFAAAA")
300 CALL CHAR(117,"5555FF818
1FF5555")
310 CALL CHAR(45,"000000FF00
000000")
320 CALL CHAR(33,"FFC3A59999
A5C3FF")
330 CALL ERKLAERUNG
340 CALL HCHAR(5,3,40,4)
350 CALL HCHAR(20,3,40,4)
360 CALL VCHAR(5,3,40,15)
370 CALL VCHAR(5,6,40,6)
380 CALL VCHAR(15,6,40,6)
390 CALL HCHAR(5,28,40,4)
400 CALL HCHAR(20,28,40,4)
410 CALL VCHAR(5,31,40,15)
420 CALL VCHAR(5,28,40,6)
430 CALL VCHAR(15,28,40,6)
440 CALL VCHAR(6,5,96,5)
450 CALL VCHAR(11,6,101,4)
460 CALL VCHAR(15,5,96,5)
470 CALL VCHAR(11,28,117,4)
480 CALL VCHAR(6,29,114,5)
490 CALL VCHAR(15,29,114,5)
500 CALL HCHAR(12,4,41,1)
510 CALL HCHAR(12,30,41,1)
520 FOR I=1 TO 40
530 RANDOMIZE
540 AA=INT(RND*23+1)
550 AB=INT(RND*31+1)
560 CALL GCHAR(AA,AB,D)
570 IF D<>32 THEN 540
580 CALL HCHAR(AA,AB,42,1)
590 NEXT I
600 CALL HCHAR(21,1,34,32)
610 CALL HCHAR(1,1,34,32)
620 CALL VCHAR(1,1,34,48)
630 CALL VCHAR(1,32,34,24)
640 CALL HCHAR(1,1,34,32)
650 A=8 :: B=8
660 CALL COLOR(3,2,11,4,2,11
,5,2,11,6,2,11,7,2,11,8,2,11
)
670 IF P=1 THEN A=16 :: B=16
680 IF P=2 THEN A=16 :: B=24
8
690 DISPLAY AT(22,1):"PANZER
";PA(1);"MINW.";MI(1);"SCHRI
TTE";SCH(1)
700 DISPLAY AT(23,1):"PANZER
";PA(2);"MINW.";MI(2);"SCHRI
TTE";SCH(2)
710 IF PA(2)=0 AND MI(2)=0 T
HEN 2190
720 IF PA(1)=0 AND PA(1)=0 T
HEN 2190
730 CALL SPRITE(#1,33,7,A+1,
B+1)
740 Z(P)=Z(P)-0.01
750 DISPLAY AT(24,1):"ZEIT:"
;Z(P)
```

```
760 IF Z(1)<0 THEN GOSUB 205
0 :: CALL DELSPRITE(#4):: Z(
1)=5.00
770 IF Z(2)<0 THEN GOSUB 205
0 :: CALL DELSPRITE(#4):: Z(
2)=5.00
780 CALL JOYST(P,X,Y)
790 IF Y=-4 THEN A=A+8
800 IF Y=4 THEN A=A-8
810 IF X=4 THEN B=B+8
820 IF X=-4 THEN B=B-8
830 IF A<8 THEN A=8
840 IF A=168 THEN A=A-8
850 IF B=8 THEN B=B+8
860 IF B=256 THEN B=B-8
870 CALL KEY(P,K,S):: IF S=0
 THEN 710 ELSE 880
880 IF K=18 THEN 900
890 GOTO 730
900 C=0 :: CALL GCHAR(A/8+1,
B/8+1,D)
910 IF D=96 AND P=1 THEN C=9
6
920 IF D=101 AND P=1 THEN C=
101
930 IF D=114 AND P=2 THEN C=
114
940 IF D=117 AND P=2 THEN C=
117
950 IF C=0 THEN 730
960 CALL DELSPRITE(#1):: CAL
L HCHAR(A/8+1,B/8+1,32)
970 IF C>101 THEN C1=2
980 IF C<102 THEN C1=5
990 CALL DELSPRITE(#2):: CAL
L SPRITE(#1,C,C1,A+1,B+1)
1000 DISPLAY AT(22,1):"PANZE
R";PA(1);"MINW.";MI(1);"SCHR
ITTE";SCH(1)
1010 DISPLAY AT(23,1):"PANZE
R";PA(2);"MINW.";MI(2);"SCHR
ITTE";SCH(2)
1020 IF SCH(1)<1 AND P=1 THE
N P=2 :: SCH(2)=4 :: CALL HC
HAR(A/8+1,B/8+1,C):: GOTO 67
0
1030 IF SCH(2)<1 AND P=2 THE
N P=1 :: SCH(1)=4 :: CALL HC
HAR(A/8+1,B/8+1,C):: GOTO 67
0
1040 CALL JOYST(P,X,Y)
1050 IF Y=-4 THEN 1060 ELSE
1120
1060 SCH(P)=SCH(P)-1 :: A=A+
8 :: CALL GCHAR(A/8+1,B/8+1,
D):: IF D=32 THEN 1120
1070 IF D=100 OR D=116 THEN
1100
1080 IF D<>32 THEN A=A-8
1090 IF D=118 THEN 1100 ELSE
 1120
```

LISTINGS

```
1100 IF C=96 OR C=114 THEN C
ALL HCHAR(A/8+1,B/8+1,42)::
PA(P)=PA(P)-1 :: CALL SOUND(
1000,-7,10):: CALL DELSPRITE
(#1):: GOTO 670
1110 IF C=101 OR C=117 THEN
CALL HCHAR(A/8+1,B/8+1,32)::
 CALL SOUND(100,330,0):: CAL
L SOUND(100,110,10):: GOTO 1
120
1120 IF Y=4 THEN 1130 ELSE 1
190
1130 SCH(P)=SCH(P)-1 :: A=A-
8 :: CALL GCHAR(A/8+1,B/8+1,
D):: IF D=32 THEN 1190
1140 IF D=100 OR D=116 THEN
1170
1150 IF D<>32 THEN A=A+8
1160 IF D=118 THEN 1170 ELSE
 1190
1170 IF C=96 OR C=114 THEN C
ALL HCHAR(A/8+1,B/8+1,42)::
PA(P)=PA(P)-1 :: CALL SOUND(
1000,-7,10):: CALL DELSPRITE
(#1):: GOTO 670
1180 IF C=101 OR C=117 THEN
CALL HCHAR(A/8+1,B/8+1,32)::
 CALL SOUND(100,330,0):: CAL
L SOUND(100,110,10):: GOTO 1
190
1190 IF X=4 THEN 1200 ELSE 1
260
1200 SCH(P)=SCH(P)-1 :: B=B+
8 :: CALL GCHAR(A/8+1,B/8+1,
D):: IF D=32 THEN 1260
1210 IF D=100 OR D=116 THEN
1240
1220 IF D<>32 THEN B=B-8
1230 IF D=118 THEN 1240 ELSE
 1260
1240 IF C=96 OR C=114 THEN C
ALL HCHAR(A/8+1,B/8+1,42)::
PA(P)=PA(P)-1 :: CALL SOUND(
1000,-7,10):: CALL DELSPRITE
(#1):: GOTO 670
1250 IF C=101 OR C=117 THEN
CALL HCHAR(A/8+1,B/8+1,32)::
 CALL SOUND(100,330,0):: CAL
L SOUND(100,110,10):: GOTO 1
260
1260 IF X=-4 THEN 1270 ELSE
1330
1270 SCH(P)=SCH(P)-1 :: B=B-
8 :: CALL GCHAR(A/8+1,B/8+1,
D):: IF D=32 THEN 1330
1280 IF D=100 OR D=116 THEN
1310
1290 IF D<>32 THEN B=B+8
1300 IF D=118 THEN 1310 ELSE
 1330
1310 IF C=96 OR C=114 THEN C
```

```
ALL HCHAR(A/8+1,B/8+1,42)::
PA(P)=PA(P)-1 :: CALL SOUND(
1000,-7,10):: CALL DELSPRITE
(#1):: GOTO 670
1320 IF C=101 OR C=117 THEN
CALL HCHAR(A/8+1,B/8+1,32)::
 CALL SOUND(100,330,0):: CAL
L SOUND(100,110,10):: GOTO 1
330
1330 CALL KEY(P,K,S):: IF K=
18 THEN 1380
1340 CALL KEY(0,KK,SS):: IF
KK=49 THEN 1350 ELSE 1360
1350 CALL HCHAR(A/8+1,B/8+1,
C):: CALL DELSPRITE(#1):: GO
TO 670
1360 IF KK=50 AND NOT(C=101
OR C=117)THEN 1640
1370 GOTO 990
1380 IF P=1 AND C=101 THEN 1
390 ELSE 1410
1390 CALL GCHAR(A/8+1,B/8,D)
:: IF D=32 THEN CALL HCHAR(A
/8+1,B/8,100):: CALL SOUND(1
000,1100,10):: SCH(P)=SCH(P)
-1 :: GOTO 990
1400 IF D<>32 THEN 990
1410 IF P=2 AND C=117 THEN 1
420 ELSE 1440
1420 CALL GCHAR(A/8+1,B/8+2,
D):: IF D=32 THEN CALL HCHAR
(A/8+1,B/8+2,116):: CALL SOU
ND(1000,1100,10):: SCH(P)=SC
H(P)-1 :: GOTO 990
1430 IF D<>32 THEN 990
1440 IF P=1 THEN B2=+8 :: W=
0 :: SCH(1)=SCH(1)-1 :: B1=B
 ELSE 1580
1450 B1=B1+B2 :: CALL GCHAR(
A/8+1,B1/8+1,D):: IF D=32 TH
EN W=W+1 :: GOTO 1560
1460 IF D=40 THEN CALL HCHAR
(A/8+1,B1/8+1,32):: CALL SOU
ND(300,-7,20):: GOTO 990
1470 IF D=42 THEN CALL HCHAR
(A/8+1,B1/8+1,32):: CALL SOU
ND(300,-7,10):: GOTO 990
1480 IF D=41 AND A>80 THEN P
A(2)=0 :: MI(2)=0 :: CALL SO
UND(1000,-7,0,110,0):: GOTO
2190
1490 IF D=41 AND A<80 THEN P
A(1)=0 :: MI(1)=0 :: CALL SO
UND(1000,-7,0,110,0):: GOTO
2190
1500 IF D=96 THEN CALL HCHAR
(A/8+1,B1/8+1,42):: CALL SOU
ND(400,-7,5):: PA(1)=PA(1)-1
 :: D=1 :: GOSUB 1590 :: GOT
O 990
1510 IF D=101 THEN CALL HCHA
R(A/8+1,B1/8+1,42):: CALL SO
UND(300,-7,5):: MI(1)=MI(1)-
1 :: D=1 :: GOSUB 1590 :: GO
TO 990
1520 IF D=114 THEN D=2 :: CA
LL HCHAR(A/8+1,B1/8+1,42)::
CALL SOUND(300,-7,0):: PA(2)
=PA(2)-1 :: GOSUB 1590 :: GO
TO 990
1530 IF D=117 THEN D=2 :: CA
LL HCHAR(A/8+1,B1/8+1,42)::
CALL SOUND(300,-7,5):: MI(2)
=MI(2)-1 :: GOSUB 1590 :: GO
TO 990
1540 IF D=100 OR D=116 THEN
CALL SOUND(100,-7,10,2200,9)
:: GOTO 990
1550 IF D=34 THEN 990
1560 CALL SPRITE(#2,45,3,A,B
1)
1570 IF W=4 THEN 990 ELSE 14
50
1580 B2=-8 :: W=0 :: SCH(2)=
SCH(2)-1 :: B1=B :: GOTO 145
0
1590 IF P=1 AND D=1 THEN DIS
PLAY AT(24,1):"DAS WAR FALSC
H"
1600 IF P=2 AND D=2 THEN DIS
PLAY AT(24,1):"DAS WAR FALSC
H"
1610 IF P=2 AND D=1 THEN DIS
PLAY AT(24,1):"BRAVO!!.TREFF
ER."
1620 IF P=1 AND D=2 THEN DIS
PLAY AT(24,1):"BRAVO!!.TREFF
ER."
1630 FOR I=1 TO 1000 :: NEXT
 I :: DISPLAY AT(24,1):"" ::
 RETURN
1640 CALL DELSPRITE(#1)
1650 CALL HCHAR(A/8+1,B/8+1,
C)
1660 IF C=96 THEN 1670 ELSE
1740
1670 P=1 :: CALL JOYST(1,X,Y
):: IF Y=-4 THEN CALL HCHAR(
A/8+1,B/8+1,99)
1680 IF Y=4 THEN CALL HCHAR(
A/8+1,B/8+1,97)
1690 IF X=-4 THEN CALL HCHAR
(A/8+1,B/8+1,98)
1700 IF X=4 THEN CALL HCHAR(
A/8+1,B/8+1,96)
1710 CALL KEY(0,K,S):: CALL
KEY(1,KK,SS):: IF KK=18 THEN
 1810
1720 IF K=49 THEN CALL HCHAR
(A/8+1,B/8+1,96):: GOTO 670
1730 GOTO 1670
1740 P=2 :: CALL JOYST(2,X,Y
```

```
):: IF Y=-4 THEN CALL HCHAR(
A/8+1,B/8+1,115)
1750 IF Y=4 THEN CALL HCHAR(
A/8+1,B/8+1,113)
1760 IF X=4 THEN CALL HCHAR(
A/8+1,B/8+1,112)
1770 IF X=-4 THEN CALL HCHAR
(A/8+1,B/8+1,114)
1780 CALL KEY(2,K,S):: CALL
KEY(0,KK,SS):: IF K=18 THEN
1810
1790 IF KK=49 THEN CALL HCHA
R(A/8+1,B/8+1,114):: GOTO 67
0
1800 GOTO 1740
1810 CALL GCHAR(A/8+1,B/8+1,
D):: CALL HCHAR(A/8+1,B/8+1,
32):: CALL SPRITE(#1,D,C1,A,
B)
1820 IF D=96 THEN C=96 :: GO
TO 1440
1830 IF D=98 THEN C=96 :: B2
=-8 :: W=0 :: SCH(1)=SCH(1)-
1 :: B1=B :: GOTO 1450
1840 IF D=99 THEN C=96 :: B2
=+8 :: W=0 :: SCH(1)=SCH(1)-
1 :: B1=A :: GOTO 1910
1850 IF D=97 THEN C=96 :: B2
=-8 :: W=0 :: SCH(1)=SCH(1)-
1 :: B1=A :: GOTO 1910
1860 IF D=114 THEN C=114 ::
GOTO 1440
1870 IF D=112 THEN C=114 ::
B2=+8 :: B1=B :: W=0 :: SCH(
2)=SCH(2)-1 :: B1=B :: GOTO
1450
1880 IF D=115 THEN C=114 ::
B2=+8 :: W=0 :: SCH(2)=SCH(2
)-1 :: B1=A :: GOTO 1910
1890 IF D=113 THEN C=114 ::
B2=-8 :: W=0 :: SCH(2)=SCH(2
)-1 :: B1=A :: GOTO 1910
1900 END
1910 CALL GCHAR(B1/8+1,B/8+1
,D):: W=W+1 :: IF D=32 THEN
2020
1920 IF D=41 AND A>80 THEN P
A(2)=0 :: MI(2)=0 :: CALL SO
UND(1000,-7,0,110,0):: GOTO
2190
1930 IF D=40 THEN CALL HCHAR
(B1/8+1,B/8+1,32):: CALL SOU
ND(300,-7,20):: GOTO 990
1940 IF D=41 AND A<80 THEN P
A(1)=0 :: MI(1)=0 :: CALL SO
UND(1000,-7,0,110,0):: GOTO
2190
1950 IF D=96 THEN D=1 :: CAL
L HCHAR(B1/8+1,B/8+1,42):: C
ALL SOUND(400,-7,5):: PA(1)=
PA(1)-1 :: GOSUB 1590 :: GOT
O 990
1960 IF D=101 THEN D=1 :: CA
LL HCHAR(B1/8+1,B/8+1,42)::
CALL SOUND(300,-7,5):: MI(1)
=MI(1)-1 :: GOSUB 1590 :: GO
TO 990
1970 IF D=114 THEN D=2 :: CA
LL HCHAR(B1/8+1,B/8+1,42)::
CALL SOUND(300,-7,0):: PA(2)
=PA(2)-1 :: GOSUB 1590 :: GO
TO 990
1980 IF D=117 THEN D=2 :: CA
LL HCHAR(B1/8+1,B/8+1,42)::
CALL SOUND(300,-7,5):: MI(2)
=MI(2)-1 :: GOSUB 1590 :: GO
TO 990
1990 IF D=100 OR D=116 THEN
CALL SOUND(100,-7,10,2200,9)
:: GOTO 990
2000 IF D=34 THEN 990
2010 IF D=42 THEN CALL HCHAR
(B1/8+1,B/8+1,32):: CALL SOU
ND(300,-7,10):: GOTO 990
2020 CALL SPRITE(#2,47,3,B1,
B):: B1=B1+B2
2030 IF W=4 THEN 990
2040 GOTO 1910
2050 RANDOMIZE :: E=INT(RND*
18+2):: F=1
2060 IF P=1 THEN P1,P2=1
2070 IF P=2 THEN P1,P2=2
2080 CALL SPRITE(#4,43,16,E*
8-7,F*8)
2090 F=F+1
2100 CALL GCHAR(E,F,D):: IF
D=32 THEN 2170
2110 IF D=96 AND P1=1 THEN C
ALL HCHAR(E,F,118):: CALL SO
UND(1000,-7,0):: PA(1)=PA(1)
-1 :: GOTO 2170
2120 IF D=101 AND P2=1 THEN
CALL HCHAR(E,F,118):: CALL S
OUND(1000,-7,0):: MI(1)=MI(1
)-1 :: GOTO 2170
2130 IF D=114 AND P1=2 THEN
CALL HCHAR(E,F,118):: CALL S
OUND(1000,-7,0):: PA(2)=PA(2
)-1 :: GOTO 2170
2140 IF D=117 AND P2=2 THEN
CALL HCHAR(E,F,118):: CALL S
OUND(1000,-7,0):: MI(2)=MI(2
)-1 :: GOTO 2170
2150 IF D=41 AND F<10 AND P1
=1 THEN P=2 :: CALL HCHAR(E,
F,118):: CALL SOUND(1000,-7,
0):: GOTO 2190
2160 IF D=41 AND F>10 AND P1
=2 THEN P=1 :: CALL HCHAR(E,
F,118):: CALL SOUND(1000,-7,
0):: GOTO 2190
2170 IF F=31 THEN RETURN
```

LISTINGS

```
2180 GOTO 2080
2190 CALL SPGET("GAMES",Q$)
:: CALL SAY(" ",SEG$(Q$,1,70)
,"OVER"):: IF P=1 AND B>80 T
HEN 2210 ELSE IF P=1 THEN 22
40
2200 IF P=2 AND B<80 THEN 22
40 ELSE IF P=2 THEN 2210
2210 CALL SAY("THIS PLAY HAD
 WON BLUE BASE")
2220 FOR I=1 TO 20 :: CALL C
OLOR(9,11,11):: FOR L=1 TO 1
0 :: NEXT L :: CALL COLOR(9,
5,11):: NEXT I
2230 GOTO 2260
2240 CALL SAY("THIS PLAY HAD
 WON BLACK BASE")
2250 FOR I=1 TO 20 :: CALL C
OLOR(11,11,11):: FOR L=1 TO
10 :: NEXT L :: CALL COLOR(1
1,2,11):: NEXT I
2260 !
2270 Q=0 :: CALL SAY("ARE U
SURE TO PLAY THIS ",SEG$(Q$,
1,70),"AGAIN")
2280 Q=Q+1 :: CALL KEY(0,K,S
)
2290 IF Q=20 THEN 2270
2300 IF K=89 OR K=121 THEN C
ALL SAY("U PLAY THIS",SEG$(Q
$,1,70),"AGAIN"):: RUN
2310 IF K=78 OR K=110 THEN 2
320 ELSE 2280
2320 CALL CLEAR :: CALL SAY(
"#GOODBYE"):: ON ERROR 2340
2330 RUN ""
2340 END
2350 SUB ERKLAERUNG
2360 CALL COLOR(3,2,11,4,2,1
1,5,2,11,6,2,11,7,2,11,8,2,1
1)
2370 CALL CLEAR
2380 A$="00000000FF000000000
00000FF0000000000000FF00000
000000000FF000000"
2390 CALL CHAR(32,SEG$(A$,1,
16)):: CALL COLOR(1,7,11)
2400 DISPLAY AT(2,9):"TANKCO
MMANDER"
2410 DISPLAY AT(4,11):"COPYR
IGHT"
2420 DISPLAY AT(6,15):"BY"
2430 DISPLAY AT(8,8):"MATTHI
AS BOSSE"
2440 DISPLAY AT(19,4):"ZEICH
ENERKLAERUNG ?"
2450 A=1
2460 CALL KEY(0,K,S)
2470 CALL CHAR(32,SEG$(A$,A,
16)):: A=A+1
2480 IF A=58 THEN A=1
```

LISTINGS

```
2490 IF K=74 OR K=106 THEN C
ALL COLOR(1,11,11):: GOTO 25
20
2500 IF K=78 OR K=110 THEN C
ALL COLOR(1,11,11):: GOTO 26
60
2510 GOTO 2460
2520 CALL CLEAR
2530 DISPLAY AT(1,1):"-----Z
EICHENERKLAEHRUNG-----"
2540 DISPLAY AT(3,1):"(...=.
...BUSCHWERK"
2550 DISPLAY AT(5,1):"*...=9
...RUINE"
2560 DISPLAY AT(7,1):").=.
...ZENTRALE SP.1 UND 2"
2570 DISPLAY AT(9,1):"`...=.
...PANZER...SP.1"
2580 DISPLAY AT(11,1):"e...=
....MINENRAEUMER SP.1"
2590 DISPLAY AT(13,1):"r...=
....PANZER...SP.2"
2600 DISPLAY AT(15,1):"u...=
....MINENRAEUMER SP.2"
2610 DISPLAY AT(17,1):"d...=
....MINE"
2620 DISPLAY AT(19,1):"t...=
....MINE"
2630 DISPLAY AT(21,1):"v...=
....BOMBENKRATER"
2640 DISPLAY AT(24,1):"BITTE
 TASTE DRUECKEN"
2650 CALL KEY(0,K,S):: IF S<
>0 THEN 2660 ELSE 2650
2660 CALL CLEAR :: SUBEND
```

Fortsetzung von Seite 110

Variablenverzeichnis:

PA() Anzahl der Panzer Spieler 1 und 2	CALL GCHAR Abtastung AA,AB,I,L,P1,P2 Gebrauchsvariablen
MI() Anzahl der Minenräumer Spieler 1 und 2	X,Y Joystickabfrage
SCH() Anzahl der noch gehbaren Schritte Spieler 1 und 2	K,S,KK,SS Tastaturabfrage
Z() Zeit Spieler 1 und 2	B1,B2 Schußvariablen
A,B Position Visiersprite (/1)	C1 Farbcode (Sprites)
P Zeigt an, welcher Spieler gerade dran ist	W Schußweite
C,D	E,F Position Flugzeug
	Q$ Sprachschatz

SNAKE

Bei diesem Spiel müssen Sie versuchen, mit einer Schlange "Äpfel" zu fressen. Zu Beginn des Spieles werden auf das Spielfeld zehn "Äpfel" gezeichnet, die Sie in einer bestimmten Zeit auffressen müssen; mit einer Schlange, die am unteren Bildrand erscheint. Sie haben

Bitte lesen Sie weiter auf Seite 120

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *        SNAKE         *
13 ! *                      *
14 ! *     Copyright by     *
15 ! *                      *
16 ! *   Carsten  Bobek     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *      Ext. Basic      *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *      5003 Bytes      *
28 ! *                      *
29 ! ************************
100 CALL CLEAR :: RANDOMIZE
110 ! ZEICHENDEF&FARBEN
120 CALL CHAR(120,"0",128,"3
C66C3C3C3C3663C",96,"3C66C3C
3C3C3663C",136,"0000001818",
114,"FF00FF00FF00FF00")
130 CALL COLOR(13,5,1,9,5,1,
14,13,1,11,9,9)
140 ! VARIABELN
150 CALL HCHAR(2,1,112,32)::
 CALL HCHAR(24,1,112,32):: C
ALL VCHAR(2,1,113,23):: CALL
 VCHAR(2,32,113,23):: GOTO 8
00
160 HSC=0
170 F=0 :: SN=3 :: BLD=1 ::
S=-7 :: P=2 :: BO=80
180 AP=10 :: HZ=110
190 DIM A(500),B(500)
200 Y=3 :: E,D=1 :: A1=17 ::
 B1=24 :: X=3
210 ON ERROR 1030
220 A(0),A(1),A(2),B(0),B(1)
,B(2)=1
230 ! SPIELFELD
240 CALL HCHAR(2,1,112,32)::
 CALL HCHAR(24,1,112,32):: C
ALL VCHAR(2,1,113,23):: CALL
 VCHAR(2,32,113,23)
250 DISPLAY AT(1,1)ERASE ALL
:USING "SC:#### HISC:#### AP
:## SN:#":STR$(SC),STR$(HSC)
,STR$(AP),STR$(SN)
260 CALL HCHAR(2,1,112,32)::
 CALL HCHAR(24,1,112,32):: C
ALL VCHAR(2,1,113,23):: CALL
 VCHAR(2,32,113,23)
270 ON BLD GOSUB 980,870,880
,890,900,920,950
280 FOR I=1 TO AP :: GOSUB 6
90 :: CALL SOUND(50,HZ,0)::
HZ=HZ+50 :: NEXT I :: HZ=110
290 CALL SPRITE(#3,113,1,9,2
49,#1,114,2,192,249,S,0,#2,1
14,2,192,1,S,0)
300 CALL SOUND(100,990,0)
310 CALL KEY(0,K,ST)
320 IF K=83 THEN 340 ELSE IF
 K=68 THEN 350
330 ON D GOTO 400,410,420,43
0
340 ON E GOTO 400,410,420,43
0
350 ON D GOTO 420,430,410,40
0
360 CALL GCHAR(B1,A1,C):: IF
 C<>32 THEN 450
370 CALL COINC(ALL,CO):: IF
CO=-1 THEN 540
380 CALL HCHAR(B1,A1,128)::
A(X)=A1 :: B(X)=B1 :: CALL H
CHAR(B(X-Y),A(X-Y),32):: X=X
+1 :: GOTO 310
390 !RICHTUNGSAENDERUNG
400 B1=B1-1 :: D=1 :: E=4 ::
 GOTO 360
410 B1=B1+1 :: D=2 :: E=3 ::
 GOTO 360
420 A1=A1+1 :: D=3 :: E=1 ::
 GOTO 360
430 A1=A1-1 :: D=4 :: E=2 ::
 GOTO 360
440 !PUNKT GEFREESSEN
450 IF C<>136 THEN 590
460 CALL SOUND(100,110,0)
470 CALL HCHAR(B1,A1,32)
480 SC=SC+P :: AP=AP-1 :: DI
SPLAY AT(1,4)SIZE(4):STR$(SC
):: DISPLAY AT(1,22)SIZE(2):
STR$(AP)
490 Y=Y+2
500 IF AP=0 THEN CALL HCHAR(
2,16,120):: CALL MOTION(#1,0
,0,#2,0,0)
510 CALL LOCATE(#1,192,249,#
2,192,1)
520 GOTO 380
530 ! 3 P DAZU
540 AP=AP+3 :: DISPLAY AT(1,
22)SIZE(2):STR$(AP)
550 FOR I=1 TO 3 :: GOSUB 69
0 :: NEXT I
560 CALL LOCATE(#1,192,249,#
2,192,1)
570 GOTO 380
580 ! NEUE SCLANGE
590 IF C=120 THEN 650
600 IF AP=0 AND C<>32 THEN 6
70
610 SN=SN-1 :: IF SN<0 THEN
740 :: DISPLAY AT(1,28)SIZE(
1):STR$(SN)
620 FOR I=210 TO 110 STEP -1
0 :: CALL SOUND(-100,I,0)::
NEXT I
630 CALL DELSPRITE(ALL):: F=
```

```
1 :: GOTO 190
640 !NEUES BILD
650 CALL HCHAR(2,16,128):: F
OR I=110 TO 1100 STEP 100 ::
 CALL SOUND(-100,I,0):: NEXT
 I
660 BLD=BLD+1 :: IF F=0 THEN
 720 ELSE IF BLD>7 THEN SN=S
N+2 :: F=0 :: SC=SC+100 :: B
LD=1 :: S=S-2 :: GOTO 180 EL
SE F=0 :: GOTO 180
670 FOR I=310 TO 210 STEP -1
0 :: CALL SOUND(-100,I,0)::
NEXT I :: F=1 :: GOTO 180
680 ! UNTERPR.PUNKTE SETZEN
690 A2=RND*20+3 :: B2=RND*29
+2 :: IF B2>15 AND B2<19 AND
 A2>20 THEN 690 ELSE CALL GC
HAR(A2,B2,C)
700 IF C<>32 THEN 690 ELSE C
ALL HCHAR(A2,B2,136):: RETUR
N
710 !BONUS
720 DISPLAY AT(24,9)BEEP SIZ
E(10):USING "BONUS:####":BO
:: SC=SC+BO :: BO=BO+20 :: F
=0 :: FOR I=1 TO 400 :: NEXT
 I :: GOTO 180
730 ! SPIELENDE
740 DISPLAY AT(24,9)SIZE(11)
:"*GAME OVER*"
750 CALL DELSPRITE(ALL)
760 FOR I=440 TO 110 STEP -1
0 :: CALL SOUND(-100,I,0)::
NEXT I
770 IF SC>HSC THEN HSC=SC ::
 GOSUB 1000 :: DISPLAY AT(1,
14)SIZE(4):STR$(HSC):: SC=0
780 FOR I=3 TO 23 :: CALL HC
HAR(I,2,32,30):: NEXT I
790 !TITELBILD
800 DISPLAY AT(8,3):"`  `  `.
.` `  `  `  `  `  `  ":"..`  `  `.
`` `.  `  `  `  `  `  `.":"..`  `  `.
..`  `  `  `  `  `  `  `":"..`  `  `
`  `  `  `  `  `  `  `  "
810 DISPLAY AT(12,3):"...`  `
..`  `  `  `  `  `  `  ":"..`  `  `.
 `  `  `  `  `  `  `  `":"..`  `  ``
`  `  `  `  `  `  `  `  "
820 DISPLAY AT(20,6):"BY CAR
STEN BOBEK"
830 SC=0
840 DISPLAY AT(24,8)BEEP SIZ
E(13):"PRESS ANY KEY"
850 CALL KEY(0,K,ST):: IF ST
=0 THEN 850 :: IF K=61 THEN
STOP ELSE 170
860 ! BILDER 1-7
870 P=3 :: CALL HCHAR(13,14,
112,5):: CALL VCHAR(8,13,112
,10):: CALL VCHAR(8,19,112,1
0):: RETURN
880 P=4 :: CALL HCHAR(15,10,
112,11):: CALL VCHAR(10,10,1
12,5):: CALL VCHAR(10,15,112
,5):: CALL VCHAR(10,20,112,5
):: RETURN
890 P=5 :: CALL HCHAR(15,8,1
12,15):: CALL VCHAR(8,8,112,
7):: CALL VCHAR(8,22,112,7):
: CALL HCHAR(8,9,112,13):: C
ALL HCHAR(8,15,32,2):: RETUR
N
900 CALL HCHAR(13,2,112,30):
: CALL VCHAR(3,15,112,21)::
CALL HCHAR(13,8,32):: CALL H
CHAR(13,24,32):: CALL HCHAR(
7,15,32)
910 CALL HCHAR(17,15,32):: P
=6 :: RETURN
920 CALL HCHAR(6,7,112,20)::
 CALL HCHAR(19,7,112,20):: C
ALL VCHAR(7,7,112,12):: CALL
 VCHAR(7,26,112,12)
930 CALL HCHAR(10,11,112,12)
:: CALL HCHAR(15,11,112,12):
: CALL VCHAR(11,11,112,4)::
CALL VCHAR(11,22,112,4)
940 CALL HCHAR(6,16,32,2)::
CALL HCHAR(15,16,32,2):: P=8
 :: RETURN
950 CALL HCHAR(7,2,112,30)::
 CALL HCHAR(13,2,112,30):: C
ALL HCHAR(19,2,112,30)
960 FOR I=8 TO 26 STEP 6 ::
CALL VCHAR(3,I,112,21):: NEX
T I :: FOR I=5 TO 29 STEP 6
:: CALL VCHAR(3,I,32,21):: N
EXT I
970 FOR I=4 TO 22 STEP 6 ::
CALL HCHAR(I,2,32,30):: NEXT
 I :: P=10 :: RETURN
980 P=2 :: RETURN
990 ! NEW HISC
1000 DISPLAY AT(19,4)BEEP:"*
*********************": :TAB(
4);"**********************" :
: DISPLAY AT(20,4):USING "*N
EW HIGH-SCORE:#####":HSC
1010 FOR I=1 TO 400 :: NEXT
I :: RETURN
1020 ! ERR UNTERPR.
1030 CALL ERR(FC,FT)
1040 IF FC<>57 THEN STOP
1050 FOR I=Y+3 TO 2*Y+5 :: A
(X-I-1)=A(I):: B(X-I-1)=B(I)
:: NEXT I
1060 FOR I=X-Y-1 TO X-1 :: C
ALL HCHAR(B(I),A(I),32):: NE
XT I
1070 ON ERROR 1030 :: X=Y ::
 RETURN 380
1080 X=Y :: RETURN 380
```

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *     ZAHLENJAGD       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *    Alex Velders      *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     5497 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CHARSET
110 CALL CLEAR :: RANDOMIZE
:: CALL MAGNIFY(2):: CALL CO
LOR(3,16,2,4,16,2,8,2,16)::
FOR Y=9 TO 12 :: CALL COLOR(
Y,16,1):: NEXT Y
120 B$="183C7E7E7E7E3C18" ::
 CALL CHAR(64,B$):: CALL CHA
R(128,B$):: CALL CHAR(91,"18
3C7EFFFF7E3C18"):: CALL COLO
R(2,16,1)
130 A$(1)="...anfaenger " ::
 A$(2)="...erfahrene " :: A$
(3)=".....experte "
140 !
150 REM spielfeldfarben
160 !
170 CO(1)=5 :: CO(2)=13 :: C
O(3)=9 :: CO(4)=6 :: CO(5)=1
0 :: CO(6)=3
180 !
190 REM sprites-anfang
200 !
210 B=12 :: A=16 :: FOR Q=53
 TO 49 STEP -1 :: CALL SCREE
N(CO(Q-48)):: GOSUB 1170 ::
NEXT Q :: CALL DELSPRITE(ALL
):: CALL SCREEN(2)
220 DISPLAY AT(5,6):"*******
**********" :: DISPLAY AT(9,
6):"******************" :: DI
SPLAY AT(7,6):"*..zahlenjagd
...*"
230 DISPLAY AT(20,5):"spiela
nleitung..j/n" :: GOSUB 1310
240 CALL KEY(0,K,S):: IF S=0
 THEN 240 :: IF K=74 OR K=10
6 THEN 250 ELSE 320
250 DISPLAY AT(1,1)BEEP ERAS
E ALL:"schick den weissen ba
ll mit dem joystick uber die
 zah-..len 1,2,3 usw."
260 DISPLAY AT(5,1):"wenn de
r weisse ball.......schwarz
wird setzt der.....computer
 die verfolgung ein"
270 DISPLAY AT(9,1):"nachein
ander konnen sie und der com.
puter fortschritte...machen"
280 DISPLAY AT(13,1):"sie so
llen kurven machen um vorspr
ung zu behalten"
290 DISPLAY AT(16,1):"wenn d
er gong klingt ist dercomput
er eingesperrt und....verfol
gt seinen weg in ihrernahe"
300 DISPLAY AT(21,1):"es gib
t 6 runden" :: DISPLAY AT(24
,10):"<taste>"
310 CALL KEY(0,K,S):: IF S=0
 THEN 310
320 DISPLAY AT(3,4)BEEP ERAS
E ALL:"schwierigkeitsgrad"
330 DISPLAY AT(8,4):"1=";A$(
1):: DISPLAY AT(11,4):"2=";A
$(2):: DISPLAY AT(14,4):"3="
;A$(3)
340 DISPLAY AT(19,4):"taste
1,2 oder 3"
350 CALL KEY(0,K,S):: IF S=0
 THEN 350 :: IF K<49 OR K>51
 THEN 350 ELSE DIF=K-48
360 !
370 REM spielfeld aufbau
380 !
390 CALL CLEAR :: CALL SCREE
N(14):: CALL COLOR(1,1,CO(SC
+1),5,16,CO(SC+1),2,16,2)::
CALL COLOR(13,2,CO(SC+1))
400 FOR Y=9 TO 12 :: CALL CO
LOR(Y,16,CO(SC+1)):: NEXT Y
410 CALL HCHAR(1,2,42,31)::
CALL HCHAR(24,2,42,31):: CAL
L VCHAR(2,2,42,22):: CALL VC
HAR(2,32,42,22)
420 DISPLAY AT(1,5)SIZE(8):"
 runde";SC+1 :: DISPLAY AT(1
,13)SIZE(LEN(A$(DIF))):A$(DI
F):: CALL HCHAR(2,2,42,30)
430 FOR I=1 TO SC+DIF
440 X1=INT(RND*21+3):: Y1=IN
T(RND*29+3):: CALL GCHAR(X1,
Y1,I1):: IF I1=32 THEN CALL
HCHAR(X1,Y1,42)ELSE 440
450 NEXT I
460 !
470 REM zahlenbestimmung
480 !
490 FOR I=49 TO 52+SC
500 X1=INT(RND*21+3):: Y1=IN
T(RND*29+3):: CALL GCHAR(X1,
Y1,I1):: IF I1=32 THEN CALL
HCHAR(X1,Y1,I)ELSE 500
510 NEXT I
520 !
```

LISTINGS

```
530 REM startposition
540 !
550 A=INT(RND*22+6):: B=INT(
RND*14+6):: CALL GCHAR(B,A,I
1)
560 IF I1=32 THEN O=A :: P=B
 :: W=INT(RND*8)+1 :: NU=0 :
: BO=0 :: MA=0 :: TR=0 ELSE
550
570 CALL HCHAR(P,O,91):: CAL
L HCHAR(B,A,64)
580 !
590 REM joystick lenkung
600 !
610 CALL JOYST(1,X,Y):: NU=N
U+1
620 A=A+X/4 :: B=B-Y/4 :: CA
LL GCHAR(B,A,Q):: CALL HCHAR
(B,A,64):: CALL SOUND(-10,15
00,1)
630 IF Q>41 AND Q<58 THEN GO
SUB 940
640 IF NU=26-DIF*4 THEN NU=0
 :: CALL HCHAR(B,A,128):: GO
TO 680 ELSE 610
650 !
660 REM computer lenkung
670 !
680 CALL HCHAR(P,O,91):: CAL
L HCHAR(P,O,32):: CALL SOUND
(-10,500,1):: IF P=B AND O=A
 THEN 890
690 ON W GOTO 700,720,730,75
0,760,780,830,850
700 TR=TR+1 :: CALL GCHAR(P+
1,O,U):: IF U=64 OR U=128 TH
EN P=P+1 :: W=1 :: TR=0 :: G
OTO 680 ELSE 710
710 IF TR>7 THEN 1230 ELSE 7
20
720 TR=TR+1 :: CALL GCHAR(P+
1,O+1,U):: IF U=64 OR U=128
THEN P=P+1 :: O=O+1 :: W=2 :
: TR=0 :: GOTO 680 ELSE 730
730 TR=TR+1 :: CALL GCHAR(P,
O+1,U):: IF U=64 OR U=128 TH
EN O=O+1 :: W=3 :: TR=0 :: G
OTO 680 ELSE 740
740 IF TR>7 THEN 1230 ELSE 7
50
750 TR=TR+1 :: CALL GCHAR(P-
1,O+1,U):: IF U=64 OR U=128
THEN P=P-1 :: O=O+1 :: W=4 :
: TR=0 :: GOTO 680 ELSE 760
760 TR=TR+1 :: CALL GCHAR(P-
1,O,U):: IF U=64 OR U=128 TH
EN P=P-1 :: W=5 :: TR=0 :: G
OTO 680 ELSE 770
770 IF TR>7 THEN 1230 ELSE 7
80
780 TR=TR+1 :: CALL GCHAR(P-
1,O-1,U):: IF U=64 OR U=128
THEN P=P-1 :: O=O-1 :: W=6 :
: TR=0 :: GOTO 680 ELSE 820
790 !
800 REM kurvenzahler
810 !
820 BO=BO+1 :: IF BO=3 THEN
BO=0 :: W=7 :: GOTO 570 ELSE
 830
830 TR=TR+1 :: CALL GCHAR(P,
O-1,U):: IF U=64 OR U=128 TH
EN O=O-1 :: W=7 :: TR=0 :: G
OTO 680 ELSE 840
840 IF TR>7 THEN 1230 ELSE 8
50
850 TR=TR+1 :: CALL GCHAR(P+
1,O-1,U):: IF U=64 OR U=128
THEN P=P+1 :: O=O-1 :: W=8 :
: TR=0 :: GOTO 680 ELSE 700
860 !
870 REM gefasst
880 !
890 CALL SOUND(500,220,5,262
,5,330,5):: DISPLAY AT(12,2)
:" versuchen sie es nochmals
" :: GOSUB 1270
900 SC=SC-1 :: IF SC<0 THEN
SC=0 :: GOTO 390 ELSE 390
910 !
920 REM detektion grenz;
        punktenzahler
930 !
940 IF Q=42 THEN 1010 ELSE G
E=Q-48 :: MA=MA+1 :: IF GE=M
A THEN GOSUB 1170 :: GOTO 95
0 ELSE 1000
950 CALL DELSPRITE(ALL):: IF
 MA=4+SC THEN 990 ELSE RETUR
N
960 !
970 REM rundenzahler
980 !
990 SC=SC+1 :: GOSUB 1050 ::
 MA=0 :: IF SC=6 THEN 1100 E
LSE GOTO 390
1000 CALL SOUND(200,-3,1)::
DISPLAY AT(12,3):"  reihenf
olge falsch" :: GOSUB 1270 :
: MA=0 :: GOTO 390
1010 CALL SOUND(200,-6,1)::
DISPLAY AT(12,3):".....nein
 nicht so" :: GOSUB 1270 ::
MA=0 :: GOTO 390
1020 !
1030 REM subroutine positiv
          runde resultat
1040 !
1050 FOR J=1 TO SC :: FOR I=
200 TO 900 STEP 20*SC :: CAL
L SOUND(-100,I,0,1100-I,0)
1060 NEXT I :: NEXT J :: IF
```

```
SC=6 THEN RETURN ELSE DISPLA
Y AT(12,2):" bravo jetzt kom
mt runde";SC+1 :: GOSUB 1270
 :: RETURN
1070 !
1080 REM spielende
1090 !
1100 CALL CLEAR :: CALL COLO
R(1,2,2):: FOR Y=9 TO 12 ::
CALL COLOR(Y,16,2):: NEXT Y
:: DISPLAY AT(4,8)ERASE ALL:
"gratuliere "
1110 DISPLAY AT(8,5):A$(DIF)
:: DISPLAY AT(14,3):"sie hab
en es geschafft"
1120 DISPLAY AT(22,4):"...no
chmals  j/n" :: FOR I=1 TO 6
 :: CALL SCREEN(CO(I)):: FOR
 G=1 TO 10 :: NEXT G :: NEXT
 I
1130 CALL KEY(0,K,S):: IF S=
0 THEN 1120 :: IF K=106 THEN
 SC=0 :: GOTO 100 ELSE END
1140 !
1150 REM subr. zahlpassage
1160 !
1170 BS=B*8-12 :: AS=A*8-10
1180 CALL SOUND(300,523,5,65
9,5,784,5):: CALL SPRITE(#1,
Q,16,BS,AS,-8,0):: CALL SPRI
TE(#2,Q,2,BS,AS,0,8)
1190 CALL SPRITE(#3,Q,16,BS,
AS,8,0):: CALL SPRITE(#4,Q,2
,BS,AS,0,-8):: FOR I=1 TO 10
0 :: NEXT I :: RETURN
1200 !
1210 REM comp. eingesperrt
1220 !
1230 TR=0 :: BO=0 :: P=B ::
O=A :: W=INT(RND*8)+1 :: GOS
UB 1310 :: GOTO 570
1240 !
1250 REM subroutine wischen
1260 !
1270 CALL HCHAR(11,3,32,29):
: CALL HCHAR(13,3,32,29):: F
OR I=1 TO 300 :: NEXT I :: R
ETURN
1280 !
1290 REM subroutine gong
1300 !
1310 FOR GA=1 TO 2 :: FOR GB
=1 TO 10 STEP 2 :: CALL SOUN
D(-100,400,GB):: NEXT GB
1320 FOR GC=1 TO 20 STEP 2 :
: CALL SOUND(-100,200,GC)::
NEXT GC :: NEXT GA :: RETURN
```

LISTINGS

Fortsetzung von Seite 115

für jeden "Apfel" so lange Zeit, bis die Balken an den Seiten am oberen Bildrand sind. Haben Sie in dieser Zeit keinen "Apfel" mit der Schlange gefressen, werden drei neue "Äpfel" auf das Spielfeld gezeichnet. Bei jedem "Apfel" der gefressen wird, wird die Schlange um zwei Glieder länger. Für jeden Apfel den die Schlange frißt, bekommen Sie je nach Bild (s. unten) Punkte. Wenn alle Punkte aufgefressen sind müssen Sie versuchen, die Schlange durch das Loch in der oberen Spielfeldumrandung herauszufahren, um in ein anderes Bild zu kommen.
Wenn man die Runde ohne Verlust einer Schlange geschafft hat, bekommt man einen Bonus, der sich bei jeder Runde um 20 Punkte erhöht. Sobald man gegen eine Wand oder gegen die Schlange fährt, beginnt die Runde von vorne, und man verliert eine Schlange, Sind alle drei Schlangen verloren worden, ist das Spiel zu Ende, und das Titelbild erscheint wieder. Hat man alle sieben Bilder durch, beginnt das Spiel wieder bei Bild 1.
In der 1. Zeile wird angezeigt:
SC: Punkte
HSC: Höchstpunktzahl
AP: Noch zu fressende Äpfel
SN: Noch zur Verfügung stehende Schlangen

Tastenfunktionen:

>D< = 90°-Drehung der Schlange nach links
>S< = 90°-Drehung der Schlange nach rechts (jeweils von der momentanen Richtung der Schlange)
>=< = um aufzuhören (nur bei Titelbild möglich!)

Bild	1	2	3	4	5	6	7
Punkte/ Apfel	2	3	4	5	6	8	10

Programmaufbau:

130–200
Zeichen- und Farbdefinitionen
210–280
Variablen
290–430
Spielfeld; Tastaturabfrage
440–710
Richtungsänderung, Überprüfung auf Zusammenstoß; diverse Unterprogramme
720–770
Spielende
780–850
Titelbild
860–1000
Unterprogramme zum Zeichnen der einzelnen Bilder
1010–1070
Error-Unterprogramm
1080–1230
Spielanweisungen

Carsten Bobek

ZAHLENJAGD

Das Programm ist für den TI 99/4A mit Extended Basic und Joystick geschrieben. Nach dem Laden und RUN erscheint das Titelbild. Sie können mit der Taste "J" eine Spielanleitung abrufen. Durch Niederdrücken einer beliebigen Taste erscheint der Schwierigkeitsgrad. Jetzt fängt das Spiel an. Die "Alpha lock"-Taste muß gelöst werden. Oben im Spielfeld erscheint der Schwierigkeitsgrad und die Rundennummer. Abhängig von der Runde erscheinen bis zu 9 Zahlen im Spielfeld. Sie sollen mit dem Joystick 1 den weißen Ball über die Zahlen in Reihenfolge 1, 2, 3 usw. lenken. Falsche Reihenfolge, Grenzüberschreitung und Zusammenstoß mit Hindernissen sind nicht gestattet. Sie können nur Fortschritte machen, so-

Bitte lesen Sie weiter auf Seite 125

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *       DIADEM         *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *  Arnd Schmilgeit     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    11858 Bytes       *
28 ! *                      *
29 ! ************************
100 RANDOMIZE
110 CALL C
120 DIM TOT$(11),EB(5):: ST=
1
130 CALL CHAR(96,"0000030301
03071B2303030302040400000000
000080C0A0A090808080402030")
140 CALL CHAR(104,"000000000
0010305050901010102040600000C
0C080C0E0DCC0C0C0C040202030"
)
150 CALL CHAR(116,"00"&RPT$(
"01",11)&"0E0800000488902020C0
C0C0C0C0C0402010080C00")
160 CALL CHAR(108,"201109040
303030303030204081030000"&R
PT$("80",11)&"701000")
170 CALL CHAR(124,"000405050
403030303030302020200000000
08888F0C0C0C0C0C04040404040"
)
180 CALL CHAR(100,"000001010
1070B1313030303020202060000
08080E0D0C8C8C0C0C040404060"
)
190 CALL CHAR(92,"0000011111
0F0303030303020202020020A0
A020C0C0C0C0C0C04040400000")
200 CALL CHAR(112,"040405050
503030303030303020202062020A
0A0A0C0C0C0C0C0C0C040404060"
)
210 CALL CHAR(120,"F92222212
020000080710B0F1E1F3BEF9F444
4840404C0C086C262F2DAFEDFF8"
)
220 G$=RPT$("0",19)&"609C971
81"&RPT$("0",25)&"C121110A04
0"
230 CALL CHAR(38,"5555FFAAAA
AAAAFF")
240 CALL CHAR(40,"607F7F6060
7F7F6006FEFE0606FEFE06")
250 H$=RPT$("0",17)&"4020907
09020408"&RPT$("0",16)&"10A0
C8F0C8201008"
260 CALL CHAR(39,"FFCC"&RPT$
("33CC",3))
270 PRINT " DD..I..AA..DD..E
EEE M...M...D D I A..A D D E
....MM MM...D D I AAAA D D E
EEE M M M...D D I A..A D D E
....M...M"
280 PRINT " DD..I A..A DD..E
EEE M...M";:::::
290 PRINT "VON ARND SCHMILGE
IT.............BORSTELWEG 42
..............2400 LUEBECK
14"
300 PRINT :::"TEL.: 0451/396
381":::::
310 PRINT "@ 7.7.1985 !0 UHR
 26":::
320 CALL W :: CALL C :: PRIN
T ">ALPHA LOCK< LOESEN.....
...STICK 1 BENUTZEN":: :: C
ALL W
330 GOTO 1730
340 ON ST GOTO 620,850,1130,
1410,1480
350 IF S=0 THEN CALL JOYST(1
,XB,YB):: CALL KEY(1,QQ,Q)::
 IF Q AND TT=0 THEN M=-4 ::
X=X+3/4*XB :: YB=-YB*(XB=0)
360 TT=S :: CALL COINC(ALL,Q
):: IF Q THEN 1800
370 IF YB<>0 THEN 1890
380 CALL GCHAR(Y/8+3,(X+3)/8
+1-.2*(XB=4),D):: IF D=32 AN
D S=0 THEN 2010
390 CALL COINC(ALL,Q):: IF Q
 THEN 1800
400 CALL PATTERN(#5,100+XB-1
2*(S<>0)):: X=X+XB/2 :: Y=Y+
M :: S=S+M :: M=-(M+1)*(S<>0
)
410 CALL COINC(ALL,Q):: IF Q
 THEN 1800
420 CALL LOCATE(#5,Y,X)
430 RETURN
440 CALL C :: AUS=1 :: XB,YB
,S,M=0 :: DISPLAY AT(1,4):"S
TUFE";ST;"  ";LE-LW;"LEBEN"
:: CALL HCHAR(24,3,39,28)::
CALL SCREEN(12)
450 ON ERROR 460 :: RETURN
460 URS$="NFEHLTRITT" :: AUS
=1 :: GOTO 1800
470 CALL PATTERN(#5,104):: C
ALL CHAR(108,"0000000000030F
1F3E3C70604"&RPT$("0",14)&"F
FFFFFFC0000000000303033F")
480 CALL CHAR(112,"4060787C3
E1F0F0703"&RPT$("0",14)&"3E3
E3F0303C3FBFFFF3F0F")
```

LISTINGS

```
490 CALL CHAR(116,"000000F0F
FFFFF03000000000C0C0C0FC00000
00080E0F8FC3C0E070301")
500 CALL CHAR(120,"7C7CFCC0C
0C3DFFFFFFEF0000000000003070
71E7CF8F0E0C00000")
510 CALL CHAR(128,"000000008
08080C0000000010101010103"&RPT
$("0",16))::  FOR A=6 TO 24 :
: CALL DELSPRITE(#A):: NEXT
A
520 FOR A=1 TO 2 :: FOR B=1
TO 2 :: QQQQ=QQQQ+4 :: CALL
SPRITE(#A*2+B+10,104+QQQQ,-5
*(B=1)-11*(B=2),8-16*(B=2),3
2-16*(A=2))::  NEXT B
530 NEXT A
540 FOR A=5 TO 8 :: CALL VCH
AR(2,A,32,4):: NEXT A :: CAL
L COLOR(13,11,1)
550 CALL HCHAR(3,5,128):: CA
LL HCHAR(3,8,129)
560 CALL CHAR(132,"C0C0"&RPT
$("0",60))::  CALL SPRITE(#20
,132,2,24,47)
570 DISPLAY AT(10,4):"UND FU
ER DIESEN SCHUND"
580 DISPLAY AT(11,4):"MUSSTE
N ZICH VIDIO-...."
590 DISPLAY AT(12,4):"MAENNC
HEN IHR LEBEN..."
600 DISPLAY AT(13,4):"LASSEN
 ???..PFUI !!!! "
610 FOR A=1 TO 75 :: CALL CO
LOR(#20,INT(RND*16)+1):: CAL
L SOUND(-100,200+200*RND,4,5
10+200*RND,5):: NEXT A :: ST
=ST+1 :: GOTO 340
620 DISPLAY AT(10,5):"STUFE
1","...SCHLUESSEL SAMMELN !"
630 CALL W :: GOSUB 440 :: Z
A=0 :: RESTORE 760 :: URS$="
NSCHLANGENBISS"
640 CALL COLOR(1,10,1)
650 DISPLAY AT(6,1):"'..''''
'' '''' ''' '''''' ''" :: EB
(1)=24
660 DISPLAY AT(12,1):"'..'''
' '''' ''..''''' ''..." :: E
B(2)=72
670 DISPLAY AT(18,1):"..'''
''' ''' '''..'' ''''.." :: E
B(4)=120
680 CALL V(4,4,9)
690 CALL V(10,27,14):: X=32
:: Y=168 :: FOR A=6 TO 18
700 READ Q,QQ :: CALL SPRITE
(#A,132,8,Q*8+2,QQ*16):: NEX
T A
710 CALL SPRITE(#1,128,13,16
8,200,0,-4)
720 CALL SPRITE(#5,100,2,Y,X
)
730 CALL CHAR(128,G$,132,"07
040407"&RPT$("01",9)&"000000
E02020E0808080808080F0E0F")
740 GOSUB 350
750 AUS=2+(Y>150):: GOTO 740
760 DATA 2,2.5,2,10,4,14,2,6
,8,7,8,4,10,9,9,12,14,1.5,14
,7,14,9,16,12
770 DATA 14,5
780 ZA=ZA+1 :: CALL SOUND(10
0,262,5,349,5,466,5):: FOR A
=3 TO 20 :: IF A=5 THEN A=6
790 CALL COINC(#5,#A,16,Q)::
 IF Q THEN CALL DELSPRITE(#A
):: CALL SOUND(100,550,5,111
0,5):: A=27
800 NEXT A :: IF ZA<13 THEN
360
810 CALL TR
820 DISPLAY AT(10,2):"ERSTE
STUFE UEBERWUNDEN" :: CALL M
OTION(#5,-4,0)
830 CALL POSITION(#5,Y,X)::
IF Y>170 THEN CALL C :: ST=S
T+1 :: GOTO 340
840 GOTO 830
850 W=6 :: CALL C :: DISPLAY
 AT(10,5):"STUFE 2","..MESSE
RWERFER FANGEN !"
860 CALL W :: GOSUB 440 :: U
RS$=" FALLENDE KLINGE"
870 CALL COLOR(1,5,1)
880 DISPLAY AT(13,1):"&&&&&&
..&&&&..&&&&&..&&&&"
890 DISPLAY AT(19,1):"&&&&&
&&&&& &&&&& &&&&&&&&" :: EB(
1)=128 :: EB(2)=80 :: EB(3)=
48
900 CALL CHAR(128,"101010FE"
&RPT$("38",10)&"1"&RPT$("0",
35))
910 CALL HCHAR(9,3,38,25)::
CALL V(7,3,8):: CALL V(11,27
,9):: CALL V(17,4,7)
920 CALL HCHAR(9,30,39,2)::
CALL SPRITE(#4,96,2,48,232)
930 X=230 :: Y=168 :: CALL S
PRITE(#5,100,2,Y,X)
940 FOR A=7 TO 17 :: CALL SP
RITE(#A,128,6,1,A*16-56,RND*
5,0):: NEXT A
950 URS$=" STICHWUNDE"
960 GOSUB 350 :: I=-(Y<129)+
2*(Y<81):: X=X+I :: IF Y=48
THEN 1000
970 GOSUB 350 :: X=X+I
980 CALL MOTION(#INT(RND*11)
+7,RND*5,0)
990 GOTO 960
```

```
1000 FOR A=6 TO 20 :: CALL D
ELSPRITE(#A):: NEXT A :: CAL
L COLOR(#4,7):: URS$="NMESSE
RWERFER"
1010 CALL CHAR(128,"00003F7F
3F"&RPT$("0",23)&"808F8FFF80
8080000")
1020 CALL SPRITE(#1,128,2,40
,220,0,-12):: CALL SPRITE(#2
,128,6,56,200):: MX=200
1030 X=X+1 :: IF X>200 THEN
1080
1040 MX=MX-8 :: IF MX<32 THE
N CALL PATTERN(#4,108):: CAL
L PATTERN(#4,96):: GOSUB 350
 :: MX=200
1050 CALL LOCATE(#2,56,MX)
1060 GOSUB 350
1070 GOTO 1030
1080 CALL DELSPRITE(#1,#2)::
 CALL COLOR(#4,4):: AUS=3
1090 GOSUB 350 :: X=X+1 :: G
OTO 1090
1100 DISPLAY AT(10,3):"ZWEIT
E STUFE UEBERWUNDEN" :: CALL
 TR
1110 CALL MOTION(#4,0,3):: C
ALL MOTION(#5,0,3):: DISPLAY
 AT(12,2):"UND EIN ZUSAETZLI
CHES LEBEN ERRUNGEN!!!!!" ::
 CALL TR :: LE=LE+1
1120 ST=ST+1 :: GOTO 340
1130 CALL C :: DISPLAY AT(10
,5):"STUFE 3","..OBERSTE TRE
PPE ERREICHEN !" :: CALL W :
: GOSUB 440 :: X=24 :: Y=168
 :: AUS=4
1140 DISPLAY AT(6,1):"%%%%%
%%%%%%%..%%..%%%%%%%%%" :: C
ALL CHAR(37,"42100000000802
0")
1150 EB(1)=24 :: DISPLAY AT(
12,1):"&&&&&&&&&&..&&&&&& &&&
&&& &&&" :: EB(2)=72
1160 CALL COLOR(1,5,1)
1170 DISPLAY AT(19,1):"''''
''' ''''''  '''''' '''''" ::
EB(3)=128 :: CALL V(1,3,5)::
 CALL V(4,28,11):: CALL V(10
,3,10):: CALL V(17,27,7)
1180 CALL CHAR(136,RPT$("0",
64),140,"000004141C153F2E3F0
31A0F0703010100005OFCBCA870D
8F8F8DFD8A0B")
1190 CALL CHAR(128,G$,132,H$
,136,A$&RPT$("0",18)&B$)
1200 RESTORE 1210 :: FOR A=1
 TO 12 :: READ F,G,H,J,L ::
CALL SPRITE(#A,F,G,H,16*J+16
,0,L):: NEXT A
1210 DATA 136,3,30,3,0,136,3
,30,4.75,0,136,3,30,11,0,132
,9,64,1,5,100,14,168,.5,0
1220 DATA 128,13,72,14,-6,13
2,9,128,5,4,132,9,145,3,0,13
2,9,145,6,0,132,9,145,10,0
1230 DATA 128,13,168,15,-4,1
28,13,168,10,-4
1240 CALL V(10,8,5):: CALL V
(10,15,5):: CALL V(10,24,5)
1250 GOSUB 350 :: IF Y>140 T
HEN 1250
1260 AUS=1 :: URS$="NSPINNEN
BISS" :: TOM,TOMI=4
1270 TOMI=TOMI-1 :: IF TOMI=
-5 THEN TOMI=4
1280 GOSUB 350 :: TOM=TOM+TO
MI :: CALL MOTION(#7,0,TOM):
: GOSUB 350 :: IF Y>100 THEN
 1270
1290 URS$="NSCHLANGEN- ODER
SPINNENBISS"
1300 GOSUB 350 :: X=X+1 :: I
F Y>40 THEN 1300
1310 URS$="NALPTRAUMBAUM" ::
 W1,W2=0
1320 IF X<32 THEN 1390
1330 W1=W1+.03 :: W2=W2+.03
:: GOSUB 350
1340 IF RND>W1-INT(W1)THEN C
ALL PATTERN(#1,140,#3,140)::
 W1=W1-.1 ELSE CALL PATTERN(
#1,136,#3,136)
1350 GOSUB 350 :: IF RND>W2-
INT(W2)THEN CALL PATTERN(#2,
140):: W2=W2-.1 ELSE CALL PA
TTERN(#2,136)
1360 GOSUB 350 :: CALL CHAR(
136,A$&RPT$("00",9-SS)&RPT$(
"80",SS)&B$):: GOTO 1320
1370 IF Y>160 THEN URS$="NSC
HLANGENBISS" ELSE URS$="NSPI
NNENBISS"
1380 GOTO 1810
1390 CALL LOCATE(#5,24,16)::
 CALL MOTION(#5,-4,0):: DISP
LAY AT(6,7):"UNGLAUBLICH !!!
" :: DISPLAY AT(10,2):"VORLE
TZTE STUFE GEMEISTERT"
1400 CALL TR :: CALL C :: ST
=ST+1 :: GOTO 340
1410 DISPLAY AT(10,5):"DIADE
M-STUFE","  ZUM KASTEN KLETT
ERN !" :: CALL W :: GOSUB 44
0 :: CALL V(2,3,22):: URS$="
NSPINNENBISS"
1420 DIADEM,QQQQ=-1
1430 CALL CHAR(132,"00003F7F
3F"&RPT$("0",23)&"808F8FFF80
8080000"):: CALL CHAR(128,H$
)
1440 FOR A=6 TO 22 :: CALL S
```

```
PRITE(#A,132,6,8*(A-6)+31,20
0,0,-9*RND*RND-1):: NEXT A
1450 X=228 :: Y=168 :: CALL
SPRITE(#5,100,5,Y,X):: CALL
SPRITE(#4,128,13,168,28,0,10
)
1460 FOR A=5 TO 8 :: CALL VC
HAR(2,A,39,4):: NEXT A
1470 GOSUB 350 :: GOSUB 350
:: GOTO 1470
1480 CALL C :: DISPLAY AT(10
,5):"STUFE 5","...LEBEND RUN
TER KOMMEN !" :: CALL W :: G
OSUB 440
1490 AUS=1 :: URS$="NSPINNEN
BISS" :: QQQQ,DIADEM=0
1500 CALL CHAR(128,H$):: RES
TORE 1610
1510 CALL CHAR(116,"00"&RPT$
("01",11)&"0E080000048890200
0C0C0C0C0C0402010080C00")
1520 CALL CHAR(108,"20110904
0303030303030204081030000"&
RPT$("80",11)&"701000")
1530 CALL CHAR(112,"04040505
0503030303030302020620200
A0A0A0C0C0C0C0C0C0C040404060
")
1540 CALL CHAR(120,"F9222221
2020000080710B0F1E1F3BEF9F44
44840404C0C086C262F2DAFEDFF8
")
1550 DISPLAY AT(5,1):"DIADEM
.........DIA..DEM"
1560 DISPLAY AT(9,1):"DIADEM
..DIAD.............EM"
1570 DISPLAY AT(14,1):"...D.
.IAD.......E..M"
1580 DISPLAY AT(19,1):"..DI.
.......A D.........E..M"
1590 DISPLAY AT(23,1):"..DIA
DEMDIADEMDIADEM AUSGANG"
1600 FOR A=1 TO 6 :: READ P2
,P1,P3 :: CALL V(P1-1,P2+2,P
3):: NEXT A :: FOR A=1 TO 4
:: READ P1 :: CALL V(13,P1+2
,7):: NEXT A
1610 DATA 7,4,8,13,4,8,23,4,
8,18,8,9,27,8,15,1,18,7,5,10
,15,21
1620 FOR A=1 TO 5 :: READ P1
,P2 :: P1=P1-8 :: CALL SPRIT
E(#A+5,128,14,P1,200,0,P2)::
 EB(A)=P1 :: NEXT A
1630 DATA 24,-4,56,-5,96,5,1
36,6,168,-7
1640 CALL SPRITE(#5,100,3,16
,24):: Y=16 :: X=24
1650 IF X>184 THEN IF Y>150
THEN 1670
1660 GOSUB 350 :: GOTO 1650
```

LISTINGS

```
1670 FOR A=6 TO 11 :: CALL D
ELSPRITE(#A):: NEXT A :: CAL
L MOTION(#5,0,4):: CALL TR :
: CALL C
1680 DISPLAY AT(4,2):"UNENDL
ICH UNWAHRSCHEINLICH" :: CAL
L TR :: DISPLAY AT(4,2):"SIE
 HABEN ALLES GESCHAFT" :: CA
LL TR
1690 DISPLAY AT(4,2):"WOLLEN
 SIE WEITERSPIELEN ?" :: CAL
L TR :: DISPLAY AT(4,2):"JA
?? DANN VIEL SPASS !!" :: CA
LL TR
1700 DISPLAY AT(4,1):"BEIM P
ROGRAMIEREN !!!.......ES SIN
D NOCH 798 BYTES FREI"
1710 CALL TR
1720 FOR A=1 TO 99 :: PRINT
CHR$(72)&CHR$(65)&" ";:: NEX
T A :: END
1730 CALL C :: DISPLAY AT(3,
2)BEEP:"MEISTERN SIE ALLE PR
OBLEME...RETTEN SIE DAS DIAD
EM!!" :: CALL CHAR(128,G$)
1740 IF ST=1 THEN 1770
1750 DISPLAY AT(23,1)BEEP:"N
EU BEGINNEN ODER BEI STUFE";
ST;"WEITERSPIELEN ?...W"
1760 ACCEPT AT(24,22)SIZE(-1
):Q$ :: IF Q$<>"W" THEN ST=1
1770 CALL MAGNIFY(4):: CALL
SPRITE(#1,96,15,100,139,0,-5
):: CALL SPRITE(#2,128,13,10
0,200,0,-9):: CALL SCREEN(12
)
1780 CALL TR :: CALL MAGNIFY
(3):: CALL C
1790 CALL MAGNIFY(3):: CALL
C :: LE=4 :: LW=0 :: GOTO 34
0
1800 ON AUS GOTO 1810,780,11
00,1370
1810 CALL SOUND(20,110,5,-5,
0):: FOR A=1 TO 28 :: CALL M
OTION(#A,0,0):: NEXT A :: CA
LL COLOR(#5,7)
1820 LW=LW+1 :: TOT$(LW)=URS
$&STR$(ST):: CALL SCREEN(15)
:: RESTORE 1850 :: FOR A=1 T
O 16 :: READ TON,LAEN :: CAL
L SOUND(1000/LAEN,TON,0)
1830 NEXT A :: CALL C :: IF
LW>=LE THEN 1870
1840 GOTO 340
1850 DATA 117,2,117,2,117,6,
117,2,139,2,136,-10,133,-10,
130,-10,127,-10,124,-10,120,
-10,117,-10
1860 DATA 117,2,117,5,117,2,
117,5
```

```
1870 FOR A=1 TO LE :: T$=TOT
$(A):: PRINT "MANN";A;"STARB
 DURCH EINE"&SEG$(T$,1,1),SE
G$(T$,2,LEN(T$)-2);"  IN STU
FE";VAL(SEG$(T$,LEN(T$),1))
1880 PRINT :: NEXT A :: PRIN
T :: CALL W :: GOTO 1730
1890 CALL PATTERN(#5,100)::
CALL GCHAR(Y/8+1-2*(YB=-4),X
/8+1.6,Q):: IF Q<>40 AND Q<>
41 THEN 380
1900 CALL COINC(ALL,Q):: IF
Q THEN 1800
1910 IF YB<>0 THEN MO=MO+32
:: IF MO=64 THEN MO=0
1920 CALL JOYST(1,XB,YB)
1930 CALL PATTERN(#5,92+MO):
: CALL GCHAR(Y/8+1-2*(YB=-4)
,X/8+1,Q):: IF Q<>40 AND Q<>
41 THEN IF XB=0 THEN 1900 EL
SE 1990
1940 CALL COINC(ALL,Q):: IF
Q THEN 1800
1950 Y=Y-YB :: CALL LOCATE(#
5,Y,X):: CALL JOYST(1,XB,YB)
:: IF DIADEM THEN IF Y<24 TH
EN CALL TR :: GOTO 470
1960 IF QQQQ THEN URS$=" STI
CHWUNDE" :: QQQQ=0
1970 CALL COINC(ALL,Q):: IF
Q THEN 1800
1980 IF XB=0 THEN 1910
1990 FOR A=1 TO 5 :: IF EB(A
)-3<=Y AND EB(A)+3>=Y THEN Y
B=0 :: Y=EB(A):: GOTO 380
2000 NEXT A :: GOTO 1900
2010 T1=3 :: M=0 :: CALL PAT
TERN(#5,112)
2020 M=M+1 :: X=X+XB/2 :: Y=
Y+M :: CALL LOCATE(#5,Y,X)
2030 T1=T1-.02 :: CALL SOUND
(-300,110*T1,5,110*T1+M,5,-5
,5):: IF Y<164 THEN 2020
2040 M=0
2050 CALL PATTERN(#5,120)::
CALL COLOR(#5,7):: FOR A=0 T
O 30 STEP 5 :: CALL SOUND(-1
00,110,A,121,A,132,A,-5,A)::
 NEXT A
2060 URS$="NSTURZ" :: GOTO 1
820
2070 SUB C :: CALL CLEAR ::
CALL DELSPRITE(ALL):: SUBEND
2080 SUB W :: DISPLAY AT(24,
2):"WEITER : BELIEBIGE TASTE
!!"
2090 CALL KEY(0,A,B):: IF B=
0 THEN 2090
2100 SUBEND
2110 SUB V(A,B,D):: CALL VCH
AR(A,B,48,D):: CALL VCHAR(A,
B+1,41,D):: SUBEND
2120 SUB TR :: RESTORE 2130
:: FOR A=3 TO 9 :: CALL SCRE
EN(A+1):: READ T,O,L :: CALL
 SOUND(1200/L,T,3,T*2+10,3,O
,5):: NEXT A
2130 DATA 440,220,4,294,220,
8,440,220,8,494,294,4,392,24
7,4,370,220,2,392,196,2
2140 SUBEND
```

LISTINGS

Fortsetzung von Seite 120

lange die hohen Töne erschallen. Wenn die hohen Töne aufhören, wird der weiße Ball schwarz. Dann müssen Sie machtlos zusehen, wie der Computer mit der Verfolgung einsetzt. Auf diese Weise können Sie und der Computer nacheinander Fortschritte machen. Um Vorsprung über den Computer zu gewinnen, müssen Sie mit dem Joystick Kurven machen, denn so wird der Computer aufgehalten. Ein Gong bedeutet, daß der Computer auf einer Stelle eingesperrt ist und dort nicht weiter kann. Passen Sie jetzt auf, denn der Computer verfolgt seinen Weg in Ihrer Nähe. Nach 6 Runden haben Sie es geschafft.

Programmunterteilung

310–380
Colorsets, Daten einlesen
390–560
Spielanleitung
580–780
Bildschirmaufbau
800–850
Joystick Lenkung
870–1060
Computer Lenkung
1080–1340
Eintragung Resultate
1360–1530
Subroutinen

Wichtige Variablen

CO(I)
Spielfeldfarben
DIF
Schwierigkeitsgrad
SC
Rundennummer
B, A
x,y Position weißer Ball
P, O
x,y Position Computer
NU
Laufzeit Joystick
W
Richtungs-Constante Computer
BO
Kurvenzähler
MA
Reihenfolgezahl
TR
Einsperrungsdetektor

DIADEM

Das vorliegende Spiel ist von der Art her in dieselbe Kategorie einzuordnen wie Miner 2049 und Kingkong. Es besitzt fünf verschiedene Screens, wobei einige sehr schwer zu bewältigen sind. Da heißt es Üben. Eine gewisse Erleichterung besteht darin, daß man nach Verlust aller vier Leben gefragt wird, ob man in dem zuletzt gespielten Screen weitermachen oder ganz von vorne beginnen möchte. In beiden Fällen gibt es dann vier neue Leben. Gespielt wird mit Joystick (1). Gesprungen wird mit der Feuertaste (einfach in die Höhe). Falls dabei der Steuerknüppel nach links oder rechts bewegt wird, erfolgt der Sprung in die entsprechende Richtung. Nach links, bzw. rechts und in die Höhe, bzw. nach unten geklettert wird mit den jeweiligen Bewegungen des Steuerknüppels.

GELD SPIEL-AUTOMAT

Zum Ablauf des Programms wird das Extended Basic Modul benötigt.
Bei diesem Spiel handelt es sich um eine Simulation der Geldspielgeräte, die man in Kneipen und Spielhallen findet. Der Einsatz pro Spiel beträgt 0,30 DM.

Bildschirmaufbau:

Links und rechts auf dem Bildschirm sieht man je eine Risikoleiste, die wieder in zwei Teile gegliedert ist, einen schwarzen und einen grünen. Die Ziffern im schwarzen Teil stellen Geldbeträge dar, die Ziffern im grünen Teil stehen für Sonderspiele. Der Mittelteil des Bildschirms wird von drei Walzen eingenommen, die Numerierung befindet sich am unteren Bildschirmrand.
Am oberen Bildschirmrand befinden sich noch Anzeigen für Serien und Kredit.
Sind alle Daten des Programms eingelesen, ertönt ein kurzer Ton, der Spieler kann jetzt seinen Einsatz eingeben . (Tasten 1–9) Mit 0 wird die Eingabe beendet.

Spielablauf:

Durch farbliche Veränderungen innerhalb der drei Walzen soll ein Laufen derselben symbolisiert werden. Nach ein paar Sekunden stoppt Walze 1, es erscheinen Ziffern oder Sterne. Gleichzeitig erscheint in dem Feld unter Walze 1 ein roter Punkt. Solange der Punkt leuchtet, kann Walze 1 nochmal gestartet werden. Dazu muß nur eine beliebige Taste gedrückt werden.
Walze 2 und 3 können vorzeitig gestoppt werden, genau dann, wenn unter der entsprechenden Walze der rote Punkt leuchtet. Auch zum Stoppen muß nur eine beliebige Taste gedrückt werden.

Gewinnplan:

Stehen alle drei Walzen still, zeigt der Computer an, ob und wieviel gewonnen wird.
Gewonnen wird, wenn alle drei Walzen einmal den selben Wert aufweisen. Der Gewinn wird in der entsprechenden Leiste durch einen roten Punkt angezeigt. Bei einem 0,30 oder 0,40 DM Gewinn erfolgt noch eine kleine Ausspielung in der entsprechenden Leiste, bei der die Chance besteht, einen höheren Gewinn zu erzielen.

Risiko:

Nachdem der Gewinn angezeigt worden ist, leuchten im Wechsel das Feld 0 und der nächsthöhere Gewinn auf. Durch Drücken einer Taste (außer A und T) kann man den Gewinn riskieren. Es kann solange riskiert werden, bis man entweder verloren oder 90 bzw. 100 Sonderspiele erreicht hat. Durch Drücken der Taste A übernimmt man den Gewinn. Durch Drücken der Taste T kann der Gewinn ab 5 bzw. 6 Sonderspielen halbiert werden.

Serie:

Die Anzahl der Sonderspiele wird am oberen Bildschirmrand angezeigt. In einer Serie wird jeder Gewinn auf 3 DM erhöht, sofern Walze 2 einen weißen Hintergrund hat.
Ab Sonderspiel 9 kann wieder riskiert werden.

Startautomatik:

Ein-Ausschaltung nach jedem Spiel durch Drücken der Taste S, es erscheint in Feld 1 ein S. Die Automatik bewirkt, daß Walze 1 nachgestartet wird, sofern kein Stern erscheint.

Risikoautomatik:

Ein-Ausschaltung nach jedem Spiel durch Drücken der Taste R, es erscheint in Feld 2 ein R. In der Automatik werden die Gewinne riskiert bis entweder verloren oder 2 bzw. 3 Sonderspiele erreicht sind. Während einer Serie ist die Risikoautomatik wirkungslos.

WAR-GAMES

Das Programm wird mit "RUN" gestartet. Auf dem Bildschirm erscheinen Nordamerika, ein weißer rechteckiger Raketenzielpunkt und ein rotes Fadenkreuz, das durch Joystick 1 gesteuert wird.
Nach einigen Sekunden verschwindet der weiße Raketenzielpunkt und gleichzeitig greifen zwei Raketen von links und rechts an. Die Raketen müssen durch Drücken der Feuertaste vernichtet werden, bevor eine der beiden Raketen den Zielpunkt erreicht. Wenn die Rakete ihren Zielpunkt erreicht, erscheint auf dem Bildschirm ein weißer Fleck und ein Punkteabzug zwischen 40 und 90 Punkte. Wenn alle 100 Energieeinheiten verbraucht worden sind, ist eine Runde beendet.
Es ist möglich, Freienergie zu bekommen, wenn man eine bestimmte Punkteanzahl erreicht hat.
Bonus: Man bekommt nach einer Runde 100 Extrapunkte, wenn man in dieser Runde keinen einzigen Fehler gemacht hat.

SOLO-WÜRFELN

Solo-Würfeln ist ein Würfelspiel für 1 Person. Man teilt einen Wurf (5 Würfel) in 2 Paare zu je 2 Würfeln und einem Rest aus einem Würfel. Zum Zusammenstellen der Paare drückt man die entsprechende Zahl (1–5).

Die nächste TI-REVUE erscheint am 25.April 86

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *  GELDSPIELAUTOMAT    *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! * Michael Lindenbaum   *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     9488 Bytes       *
28 ! *                      *
29 ! ************************
100 OPTION BASE 1
110 DIM LR(12,2),M(12,2),FLR
(12,2),GE(19,2)
120 RANDOMIZE :: CALL MAGNIF
Y(2)
130 CRE,SER=0 :: CALL CLEAR
:: CALL FARBEN(FLR(,),M(,))
140 CALL CHARDEF
150 CALL BILD
160 CALL WERTE(LR(,),M(,),GE
(,))
170 IF CRE<.30 THEN CALL CRE
DIT(CRE):: IF CRE<.30 THEN C
ALL CLEAR :: CALL DELSPRITE(
ALL):: CALL CHARSET :: END
180 CALL DREH(LR(,),M(,),FLR
(,),W1,W2,W3,W4,W5,SFL,STAU)
:: IF SER=0 THEN SFL=0
190 CRE=CRE-.30 :: CALL CRED
IS(CRE)
200 CALL GEWO(W1,W2,W3,W4,W5
,GEW):: IF (GEW=0)AND(SFL=0)
THEN 270
210 IF GEW=19 THEN CALL MELO
 :: CALL ALL(GEW,CRE,SER,GE(
,),37):: GOTO 280
220 IF GEW/2=INT(GEW/2)THEN
SP=197 ELSE SP=37
230 IF SER<>0 THEN CALL SER(
GEW,GE(,),CRE,SER,SP,W3):: G
OTO 280
240 FOR I=600 TO 803 STEP 20
0 :: CALL SOUND(200,I,0):: N
EXT I
250 IF (GEW=8)OR(GEW=9)OR(GE
W=1)OR(GEW=2)THEN CALL AUS(G
EW,SP,GE(,))
260 CALL RISIKO(GEW,GE(,),CR
E,SER,SP,RAU):: GOTO 290
270 CALL SOUND(300,250,0)
280 IF SER<>0 THEN SER=SER-1
 :: CALL SERDIS(SER)
290 FOR I=1 TO 300 :: NEXT I
 :: CALL PAUSE(RAU,STAU,GE(,
)):: GOTO 170
300 SUB FARBEN(FLR(,),M(,))
310 CALL SCREEN(14):: FOR I=
1 TO 11 :: CALL COLOR(I,2,15
):: NEXT I :: CALL COLOR(12,
13,15)
320 DATA 2,2,7,5,13,6,7,5,2,
10,7,6,9,5,7,10,2,14,7,5,13,
10,7,5
330 RESTORE 320 :: FOR I=1 T
O 12 :: READ A,B :: FLR(I,1)
=A :: FLR(I,2)=B :: NEXT I
340 DATA 2,5,7,6,7,10,5,9,7,
14,7,13
350 RESTORE 340 :: FOR I=1 T
O 12 :: READ A :: M(I,2)=A :
: NEXT I
360 SUBEND
370 SUB CHARDEF
380 CALL CHAR(96,"00F7151575
1515F7",97,"008785A5A5F52527
",98,"00F78585F59595F7")!30,
40,60
390 CALL CHAR(99,"00F79595F5
9595F7",100,"002F61A12F28282
F",101,"002F68A82F29292F")!8
0,1.20,1.60
400 CALL CHAR(102,"00E8282AE
A8F82E2",103,"00925438FE3854
92",104,"00003C7EFF7E3C00")!
2.40,SONNE,DISC
410 CALL CHAR(105,"002765A52
5252527",106,"00F71515F58585
F7",107,"00F71414F78181F7")!
10,20,25
420 CALL CHAR(108,"00F78585F
51515F7",109,"00F79595F51515
F7",110,"005FD5555555555F")!
50,90,100
430 A$="1818181818181818" ::
 B$="000000FFFF" :: C$="0000
001F1F181818" :: D$="000000F
8F8181818" :: E$="1818181F1F
" :: F$="181818F8F8"
440 G$="1818181F1F181818" ::
 H$="181818F8F8181818"
450 CALL CHAR(112,A$,113,B$,
114,G$,115,H$,116,E$,117,F$,
120,A$,121,B$,122,C$,123,D$,
124,G$,125,H$)!RAHMEN
460 CALL CHAR(93,"FFFFFFFFFF
FFFFFF",94,C$,95,D$)
470 SUBEND
480 SUB BILD
490 DATA "..zy{ SERIE....CRE
DIT......."
500 DATA "..xnx....0........
,00 DM...."
510 DATA "..!y}.............
....zy{..."
520 DATA "..x1x...^qq_....^q
q_..xmx..."
```

```
530 DATA "..!y}...p..p....p.
.p..!y}..."
540 DATA "..xkx...p..p....p.
.p..xax..."
550 DATA "..!y}...tqqu^qq_tq
qu..!y}..."
560 DATA "..xdx.......p..p..
....xjx..."
570 DATA "..!y}.......p..p..
....!y}..."
580 DATA "..x6x...^qq_tqqu^q
q_..xix..."
590 DATA "..!y}...p..p....p.
.p..!y}..."
600 DATA "..x3x...p..p....p.
.p..x5x..."
610 DATA "..!y}...tqqu....tq
qu..!y}..."
620 DATA "..pfp.............
....x2x..."
630 DATA "..rqs.............
....!y}..."
640 DATA "..pdp.............
....pep..."
650 DATA "..rqs....LINDESOFT
....rqs..."
660 DATA "..pbp.............
....pcp..."
670 DATA "..rqs.....DISC 85.
....rqs..."
680 DATA "..p`p.............
....pap..."
690 DATA "..rqs.............
....rqs..."
700 DATA "..p0p...^q_ ^q_..^
q_..p0p..."
710 DATA "..tqu...p1p p2p..p
3p..tqu..."
720 DATA "........tqu tqu..t
qu........."
730 RESTORE 490 :: FOR I=1 T
O 24 :: READ A$ :: DISPLAY A
T(I,1):A$ :: NEXT I
740 CALL SPRITE(#2,103,2,33,
89,#3,104,2,81,89,#4,103,2,5
7,121,#5,103,2,33,153,#6,104
,2,81,153)
750 SUBEND
760 SUB WERTE(LR(,),M(,),GE(
,))
770 DATA 3,4,-3,-4,1,0,-3,-4
,3,-2,-3,0,-1,-4,-3,-2,3,2,-
3,-4,1,-2,-3,-4
780 RESTORE 770 :: FOR I=1 T
O 12 :: READ A,B :: LR(I,1)=
A+100 :: LR(I,2)=B+100 :: NE
XT I
790 DATA 3,-4,-3,0,-3,-2,-4,
-1,-3,2,-3,1
800 RESTORE 790 :: FOR I=1 T
O 12 :: READ A :: M(I,1)=A+1
00 :: NEXT I
810 DATA .30,149,.40,149,.60
,133,.80,133,1.20,117,1.60,1
17,2.40,101,2,101,3,85,5,85
820 DATA 6,69,10,69,12,53,20
,53,25,37,40,37,50,21,90,21,
100,5
830 RESTORE 810 :: FOR I=1 T
O 19 :: READ A,B :: GE(I,1)=
A :: GE(I,2)=B :: NEXT I
840 SUBEND
850 SUB CREDIT(CRE)
860 CALL SOUND(200,500,0)
870 CALL KEY(0,K,S)
880 IF (K=48)THEN SUBEXIT
890 IF (K<49)OR(K>57)THEN 87
0 ELSE CRE=CRE+(K-48):: CALL
 CREDIS(CRE)
900 FOR I=1 TO 50 :: NEXT I
:: GOTO 860
910 SUBEND
920 SUB DREH(LR(,),M(,),FLR(
,),W1,W2,W3,W4,W5,SFL,STAU)
930 W1,W2,W3,W4,W5=93 :: F1,
F2,F3,F4,F5,F=2 :: SFL=0
940 CALL DELSPRITE(#9):: CAL
L ZUF1(ANZ):: CALL WAN(W1,W2
,W3,W4,W5):: FOR I=ANZ TO 11
+ANZ :: CALL FA(F):: CALL FA
R(F-1,F+1,F,F+1,F-1):: NEXT
I
950 W1=LR(ANZ,1):: W2=LR(ANZ
,2):: F1=FLR(ANZ,1):: F2=FLR
(ANZ,2):: CALL FAR(F1,F2,F,F
+1,F-1):: CALL WAN(W1,W2,W3,
W4,W5):: CALL SPRITE(#7,93,7
,173,85)
960 FOR I=ANZ TO ANZ+6 :: CA
LL ABFR(4,5,6,F):: CALL KEY(
0,K,S)
970 IF S<>0 THEN 990
980 NEXT I :: IF STAU=1 AND
W1<>103 THEN 990 ELSE 1000
990 W1,W2=93 :: CALL WAN(W1,
W2,W3,W4,W5)
1000 CALL LOCATE(#7,173,157)
1010 CALL ZUF1(ANZ):: FOR I=
ANZ TO 11+ANZ :: CALL FA(F)
1020 IF W1<>93 THEN CALL FAR
(F1,F2,F,F+1,F-1)ELSE CALL F
AR(F-1,F+1,F,F+1,F-1)
1030 CALL KEY(0,K,S):: IF S<
>0 THEN 1050
1040 NEXT I
1050 IF I<=12 THEN AN=I ELSE
 AN=I-12
1060 W4=LR(AN,1):: W5=LR(AN,
2):: F4=FLR(AN,1):: F5=FLR(A
N,2):: CALL FAR(F1,F2,F3,F4,
F5):: CALL WAN(W1,W2,W3,W4,W
5):: CALL DELSPRITE(#7)
```

```
1070 IF W1<>93 THEN 1110
1080 CALL ZUF1(ANZ):: FOR I=
ANZ TO 6+ANZ :: CALL ABFR(2,
3,4,F):: NEXT I
1090 IF I<=12 THEN AN=I ELSE
 AN=I-12
1100 W1=LR(AN,1):: W2=LR(AN,
2):: F1=FLR(AN,1):: F2=FLR(A
N,2):: CALL FAR(F1,F2,F3,F4,
F5):: CALL WAN(W1,W2,W3,W4,W
5)
1110 CALL SPRITE(#7,93,7,173
,117):: CALL ZUF1(ANZ):: FOR
 I=ANZ TO 11+ANZ :: CALL FA(
F):: CALL COLOR(#4,F):: CALL
 KEY(0,K,S)
1120 IF S<>0 THEN 1140
1130 NEXT I
1140 IF I<=12 THEN AN=I ELSE
 AN=I-12
1150 W3=M(AN,1):: F3=M(AN,2)
:: CALL COLOR(#4,F3):: CALL
PATTERN(#4,W3):: IF AN/2=INT
(AN/2)THEN 1160 ELSE CALL SP
RITE(#9,93,16,57,121):: SFL=
1
1160 CALL DELSPRITE(#7)
1170 SUBEND
1180 SUB GEWO(W1,W2,W3,W4,W5
,GEW)
1190 GEW=0
1200 IF W3<>103 THEN 1250
1210 IF W1+W2+W3+W4+W5=517 T
HEN GEW=19 :: SUBEXIT
1220 IF W1+W2+W3+W4=413 OR W
1+W3+W4+W5=413 THEN GEW=9 ::
 SUBEXIT
1230 IF W1+W3+W4=309 THEN GE
W=8 :: SUBEXIT
1240 GOTO 1290
1250 IF (W1<>W3)AND(W2<>W3)T
HEN GEW=0 :: SUBEXIT
1260 IF ((W1=W4)OR(W1=W5))AN
D(W1=W3)THEN GEW=W1-95 :: SU
BEXIT
1270 IF ((W2=W4)OR(W2=W5))AN
D(W2=W3)THEN GEW=W2-95 :: SU
BEXIT
1280 GEW=0 :: SUBEXIT
1290 IF (W1<>W4)AND(W1<>W5)A
ND(W2<>W4)AND(W2<>W5)THEN GE
W=1 :: SUBEXIT
1300 IF (W1=W4)AND(W2=W5)THE
N GEW=MAX(W1,W2)-95 :: SUBEX
IT
1310 IF (W1=W4)OR(W1=W5)THEN
 GEW=W1-95 :: SUBEXIT
1320 GEW=W2-95
1330 SUBEND
1340 SUB RISIKO(GEW,GE(,),CR
E,SER,SP,RAU)
1350 IF GEW>7 THEN F=5 ELSE
F=7
1360 CALL SPRITE(#7,93,F,GE(
GEW,2),SP):: FOR I=1 TO 200
:: NEXT I :: CALL DELSPRITE(
#8)
1370 Z1=165 :: Z2=GE(GEW+2,2
):: FOR I=1 TO 35 :: CALL LO
CATE(#7,Z1,SP):: CALL KEY(0,
K,S)
1380 IF S<>0 THEN 1410
1390 H=Z1 :: Z1=Z2 :: Z2=H :
: NEXT I :: K=65
1400 IF RAU=1 AND GEW<8 THEN
 1430
1410 IF (K=65)THEN CALL ALL(
GEW,CRE,SER,GE(,),SP):: SUBE
XIT
1420 IF (K=84)THEN CALL TEIL
(GEW,CRE,SER,GE(,),SP):: GOT
O 1360
1430 CALL SPRITE(#8,93,7,173
,117):: CALL PEEK(-31879,ZN)
:: IF ZN/2=INT(ZN/2)THEN 147
0
1440 CALL LOCATE(#7,GE(GEW+2
,2),SP):: FOR I=333 TO 999 S
TEP 333 :: CALL SOUND(I-222,
I,0):: NEXT I :: GEW=GEW+2
1450 IF GEW>7 THEN F=5 :: IF
 GEW>17 THEN CALL MELO :: K=
65 :: GOTO 1410
1460 GOTO 1360
1470 CALL LOCATE(#7,165,SP):
: FOR I=300 TO 150 STEP -50
:: CALL SOUND(200,I,0):: NEX
T I :: CALL DELSPRITE(#7,#8)
1480 SUBEND
1490 SUB SER(GEW,GE(,),CRE,S
ER,SP,W3)
1500 IF (GEW=8)OR(GEW=9)THEN
 CALL AUS(GEW,SP,GE(,)):: CA
LL RISIKO(GEW,GE(,),CRE,SER,
SP,0):: SUBEXIT
1510 IF (SER>9)AND(W3<>103)T
HEN CALL ALL(7,CRE,SER,GE(,)
,37):: GOTO 1540
1520 FOR I=600 TO 800 STEP 2
00 :: CALL SOUND(200,I,0)::
NEXT I
1530 CALL RISIKO(7,GE(,),CRE
,SER,37,0)
1540 CRE=CRE+.60 :: CALL CRE
DIS(CRE)
1550 SUBEND
1560 SUB AUS(GEW,SP,GE(,))
1570 CALL SPRITE(#7,93,7,GE(
GEW,2),SP)
1580 FOR I=1 TO 5 :: FOR J=G
EW TO GEW+8 STEP 2 :: CALL L
OCATE(#7,GE(J,2),SP):: CALL
```

```
KEY(0,K,S)
1590 IF S<>0 THEN 1610
1600 NEXT J :: NEXT I
1610 CALL PEEK(-31879,D):: I
F D<10 THEN F=8 ELSE IF D<30
 THEN F=6 ELSE IF D<90 THEN
F=4 ELSE IF D<170 THEN F=2 E
LSE F=0
1620 GEW=GEW+F
1630 SUBEND
1640 SUB CREDIS(CRE)
1650 DISPLAY AT(2,16):USING
"###.## DM":CRE
1660 SUBEND
1670 SUB SERDIS(SER)
1680 DISPLAY AT(2,8)SIZE(3):
USING "###":SER
1690 SUBEND
1700 SUB ZUF1(ANZ)
1710 ANZ=INT(RND*12)+1
1720 SUBEND
1730 SUB WAN(A,B,C,D,E)
1740 CALL PATTERN(#2,A,#3,B,
#4,C,#5,D,#6,E)
1750 SUBEND
1760 SUB FAR(A,B,C,D,E)
1770 CALL COLOR(#2,A,#3,B,#4
,C,#5,D,#6,E)
1780 SUBEND
1790 SUB ABFR(A,B,C,F)
1800 CALL FA(F):: CALL COLOR
(#A,F,#B,F-1,#C,F+1)
1810 SUBEND
1820 SUB ALL(GEW,CRE,SER,GE(
,),SP)
1830 CALL SPRITE(#8,93,7,173
,157):: CALL LOCATE(#7,GE(GE
W,2),SP):: FOR I=500 TO 800
STEP 100 :: CALL SOUND(200,I
,0,I+100,0,I+200,0):: NEXT I
1840 IF GEW<8 THEN CRE=CRE+G
E(GEW,1)ELSE SER=SER+GE(GEW,
1):: CRE=CRE+.30 :: CALL SER
DIS(SER)
1850 CALL CREDIS(CRE):: FOR
I=1 TO 200 :: NEXT I :: CALL
 DELSPRITE(#7,#8)
1860 SUBEND
1870 SUB TEIL(GEW,CRE,SER,GE
(,),SP)
1880 CALL SPRITE(#8,93,7,173
,85):: IF GEW<10 THEN CALL S
OUND(200,300,0):: CALL DELSP
RITE(#8):: SUBEXIT
1890 GEW=GEW-2 :: CALL LOCAT
E(#7,GE(GEW,2),SP):: IF GE(G
EW+2,1)/2<>INT(GE(GEW+2,1)/2
)THEN SER=SER+1
1900 SER=SER+GE(GEW,1):: CAL
L SERDIS(SER):: FOR I=600 TO
 800 STEP 200 :: CALL SOUND(
150,I,0):: NEXT I
1910 FOR I=1 TO 250.:: NEXT
I
1920 SUBEND
1930 SUB FA(F)
1940 F=F+1 :: IF F>15 THEN F
=2
1950 SUBEND
1960 SUB MELO
1970 DATA 2,783,2,783,2,880,
3,739,1,783,2,880,2,987,2,98
7,2,1046,3,987
1980 DATA 1,880,2,783,2,880,
2,783,2,739,4,783,2,43000,2,
1174,2,1174
1990 DATA 2,1174,3,1174,1,10
46,2,987,2,1046,2,1046,2,104
6,3,1046,1,987
2000 DATA 2,880,2,987,1,1046
,1,987,1,880,1,783,3,987,1,1
046,2,1174
2010 DATA 1,1318,1,1174,1,10
46,1,987,2,880,4,783,4,678
2020 FOR J=1 TO 2 :: RESTORE
 1970 :: FOR I=1 TO 44 :: RE
AD A,B :: CALL SOUND(A*125,B
,0,B-2,5,B+2,5):: NEXT I ::
NEXT J
2030 SUBEND
2040 SUB PAUSE(RAU,STAU,GE(,
))
2050 CALL SPRITE(#7,93,7,165
,37,#8,93,7,165,197)
2060 FOR I=2 TO 18 STEP 2 ::
 CALL LOCATE(#7,GE(I-1,2),37
,#8,GE(I,2),197)
2070 CALL KEY(0,K,S):: IF K=
83 AND STAU=1 THEN STAU=0 ::
 DISPLAY AT(23,10)SIZE(1):"1
" :: GOTO 2110
2080 IF K=82 AND RAU=1 THEN
RAU=0 :: DISPLAY AT(23,14)SI
ZE(1):"2" :: GOTO 2110
2090 IF K=82 AND RAU=0 THEN
RAU=1 :: DISPLAY AT(23,14)SI
ZE(1):"R"
2100 IF K=83 AND STAU=0 THEN
 STAU=1 :: DISPLAY AT(23,10)
SIZE(1):"S"
2110 NEXT I :: CALL DELSPRIT
E(#7,#8)
2120 SUBEND
```

LISTINGS

LESERBRIEFE UND FRAGEN SIND UNS STETS WILLKOMMEN. WIR BEANTWORTEN SIE ENTWEDER DIREKT ODER AUF DER LESERBRIEFSEITE

LISTINGS

```
10 ! ***********************
11 ! *                     *
12 ! *     WAR GAMES       *
13 ! *                     *
14 ! *    Copyright by     *
15 ! *                     *
16 ! *   Yuan-Tseng Lee    *
17 ! *                     *
19 ! * Benoetigte Geraete  *
20 ! *  TI99/4A  Konsole   *
21 ! *     Ext. Basic      *
22 ! *     Joystick 1      *
23 ! *                     *
26 ! *  Speicherbelegung   *
27 ! *     7660 Bytes      *
28 ! *                     *
29 ! ***********************
100 RANDOMIZE :: DIM AL(24),
TON(8)
110 E=100 :: F=1000 :: QU,SC
,Z=0 :: HI=1000 :: BO=0
120 TON(1)=247 :: TON(2)=262
 :: TON(3)=294 :: TON(4)=330
 :: TON(5)=370 :: TON(6)=392
 :: TON(7)=440 :: TON(8)=494
130 CALL SCREEN(2):: CALL CL
EAR
140 DATA 7E4242425A5A427E,30
10101010103C3C,7C04047C60606
07C,780808780C0C0C7C,4444444
4447C0C0C,7C40407C0C0C0C7C
150 DATA 7E40407E6262627E,7E
06060606060606,3C24247E62626
27E,7E42427E0606067E,0018180
000181800
160 DATA 3E22223E62626262,7C
64647E6262627E,7E46464040424
27E,7C6262626262627C,7E40407
E6060607E,7E60607E20202020
170 DATA 7E6666606E62627E,44
44447C66666666,0808080818181
818,040404040464647C,4448507
E66666666,202020206060607E
180 DATA 86CEB6868686B6B6,42
62524A46425A5A,7E42424246464
67E,7E42427E60606060,7C44444
4646C647E,7C44447E62626262
190 DATA 7E42407E0262627E,7E
18181818181818,4646464646464
67E,8686868686442810,DADAC2C
2C2D2EAC6,4242241818246262
200 DATA 8686864428101010,7E
4244081026467E
210 DATA 101088E89880804,204
080E010100C03,00102824242848
5,201010202020408,0002060A0A
0A0908
220 DATA 902020404080808,010
2020201010101,10102020404080
8,0101020202010101,808040404
080808
230 DATA 0101020202020101,02
02020202040404,8080804040404
02,2020201010100808,08040404
02020101
240 DATA 0101020204040808,80
9068080F010101,8080809894545
454,001E11302020408,00000080
700C0301
250 DATA 54522A2A2A2A2915,01
01010202020202,E010080804040
402,151515140A0A0A0A,0000008
08040402,0202010101,20202020
20C
260 DATA FF818181818181FF,E7
818181818181E7,0000C4FFC4,00
0023FF23,,
270 DATA 00000000000000FF,80
80808080808080,0101010101010
101,FF,,,,FF818181818181FF
280 FOR I=0 TO 10 :: READ C$
 :: CALL CHAR(48+I,C$):: NEX
T I
290 FOR I=0 TO 25 :: READ C$
 :: CALL CHAR(65+I,C$):: NEX
T I
300 FOR I=0 TO 40 :: READ C$
 :: CALL CHAR(96+I,C$):: NEX
T I
310 CALL CHAR(61,"0000007E00
E7",42,"7EC3A59999A5C37E")
320 DISPLAY AT(18,5):"COPYRI
GHT BY" :: DISPLAY AT(20,5):
"YUANTSENG LEE"
330 DATA 23,1,18,7,1,13,5,19
340 DATA 3,3,1,2,1,1,1,3,1,6
,1,7,4,8,4,6,3,4,1,5,1,6,1,5
,1,6,1,4,4,3,4,1
350 DATA 3,3,1,2,1,1,1,3,1,6
,1,7,4,8,2,6,1,6,1,6,2,7,2,7
,1,6,1,6,2,5,8,6,0,0
360 FOR I=5 TO 8 :: CALL COL
OR(I,5,2):: NEXT I :: CALL C
OLOR(2,13,2,3,5,2,4,14,2)
370 FOR I=1 TO 200 :: NEXT I
380 FOR A=8 TO 12 STEP 2 ::
READ AL(A):: FOR I=1 TO 26 :
: CALL HCHAR(5,A,I+64):: CAL
L SOUND(10,110,15,-5,0)
390 IF I=AL(A)THEN 410
400 NEXT I
410 NEXT A
420 FOR A=16 TO 24 STEP 2 ::
 READ AL(A):: FOR I=1 TO 26
:: CALL HCHAR(5,A,I+64):: CA
LL SOUND(10,110,15,-5,0)
430 IF I=AL(A)THEN 450
440 NEXT I
450 NEXT A
460 CALL HCHAR(6,7,61,19)
470 DISPLAY AT(10,5):"DAS SP
IEL BEGINNT IN" :: FOR I=5 T
```

```
0 0 STEP -1 :: CALL SOUND(10
,-1,0):: DISPLAY AT(12,4)SIZ
E(7):I;"SEC" :: FOR T=1 TO 1
85 :: NEXT T :: NEXT I
480 CALL HCHAR(1,3,42,28)::
CALL HCHAR(24,3,42,28):: CAL
L VCHAR(2,3,42,22):: CALL VC
HAR(2,30,42,22)
490 DISPLAY AT(10,5)SIZE(20)
:"DRUECKEN SIE BITTE" :: DIS
PLAY AT(12,5)SIZE(15):"DIE F
EUERTASTE"
500 RESTORE 340
510 READ D,F :: IF D=0 THEN
500 :: CALL KEY(1,K,S):: IF
K=18 THEN 530 :: CALL SOUND(
D*200,TON(F),0,INT(TON(F)/2)
,8)
520 GOTO 510
530 CALL CLEAR
540 FOR I=2 TO 14 :: CALL CO
LOR(I,2,2):: NEXT I
550 B,C=2 :: GE=5
560 DISPLAY AT(2,10):"`.....
....ab"
570 DISPLAY AT(3,10):"c.....
....de"
580 DISPLAY AT(4,9):"f......
....g"
590 DISPLAY AT(5,9):"f......
...h"
600 DISPLAY AT(6,10):"i.....
....i"
610 DISPLAY AT(7,9):"j......
...k"
620 DISPLAY AT(8,10):"l.....
...k"
630 DISPLAY AT(9,10):"m.....
...n"
640 DISPLAY AT(10,10):"n....
....o"
650 DISPLAY AT(11,11):"p....
...g"
660 DISPLAY AT(12,12):"q...r
sh"
670 DISPLAY AT(13,12):"t..u.
.vl"
680 DISPLAY AT(14,12):"wx k.
.yz"
690 FOR I=17 TO 23 :: CALL H
CHAR(I,4,123,5):: NEXT I
700 DISPLAY AT(18,10)SIZE(17
):"ENERGIE:"
710 DISPLAY AT(20,10)SIZE(17
):"HIGHSCORE:"
720 DISPLAY AT(22,10)SIZE(17
):"SCORE:"
730 CALL HCHAR(1,4,129,27)::
 CALL HCHAR(15,4,132,27):: C
ALL HCHAR(16,10,129,21):: CA
LL HCHAR(24,10,132,21)
```

LISTINGS

```
740 CALL VCHAR(2,4,130,13)::
 CALL VCHAR(2,30,131,13):: C
ALL VCHAR(17,10,130,7):: CAL
L VCHAR(17,30,131,7)
750 FOR I=1 TO 20 :: CALL HC
HAR(INT(RND*7)+17,INT(RND*5)
+4,136):: NEXT I
760 DISPLAY AT(18,18)SIZE(9)
:E
770 DISPLAY AT(20,20)SIZE(5)
:HI
780 DISPLAY AT(22,16)SIZE(12
):SC
790 FOR I=2 TO 8 :: CALL COL
OR(I,7,1):: NEXT I :: FOR I=
9 TO 12 :: CALL COLOR(I,5,1)
:: NEXT I :: CALL COLOR(13,1
3,1,14,5,16)
800 CALL SPRITE(#1,124,7,57,
125)
810 Z3=INT(RND*11)+2 :: S3=I
NT(RND*9)+11 :: M=0 :: GE=GE
+1 :: IF GE>12 THEN GE=12
820 GOSUB 1020
830 CALL SPRITE(#3,123,16,Z3
*8+1,S3*8+1):: FOR I=1 TO 10
 :: GOSUB 1020 :: NEXT I ::
CALL COLOR(#3,1)
840 A=INT(RND*2)+1 :: ON A G
OTO 850,860
850 CALL SPRITE(#2,125,14,Z3
*8+1,25,0,GE):: CALL SPRITE(
#4,126,14,Z3*8+1,236,0,-GE+4
):: GOTO 870
860 CALL SPRITE(#2,126,14,Z3
*8+1,236,0,-GE):: CALL SPRIT
E(#4,125,14,Z3*8+1,25,0,GE-4
)
870 CALL COINC(#2,#3,8,C2)::
 CALL COINC(#3,#4,8,C3):: IF
 (C2=-1)+(C3=-1)THEN 880 ELS
E 910
880 CALL DELSPRITE(#2,#3,#4)
:: CALL MOTION(#1,0,0):: CAL
L SOUND(-100,-7,0)
890 CALL HCHAR(Z3+1,S3+1,137
):: Z=INT(RND*40)+50 :: B0=1
900 SC=SC-Z :: DISPLAY AT(22
,16)SIZE(12):SC :: IF SC<-20
0 THEN 1130 ELSE 810
910 CALL KEY(1,K,S):: IF K<>
18 THEN GOSUB 1020 :: GOTO 8
70
920 CALL MOTION(#1,0,0):: CA
LL COINC(#1,#2,8,C1):: CALL
COINC(#1,#4,8,C4)
930 IF C1<>0 THEN 960
940 IF C4<>0 THEN 990
950 GOSUB 1020 :: E=E-5 :: D
ISPLAY AT(18,18)SIZE(9):E ::
 IF E<5 THEN 1070 ELSE 870
```

```
960 CALL DELSPRITE(#2):: CAL
L SOUND(-100,880,0):: SC=SC+
50 :: DISPLAY AT(22,16)SIZE(
12):SC :: E=E-5 :: DISPLAY A
T(18,18)SIZE(9):E
970 IF E<5 THEN 1070
980 HI=MAX(HI,SC):: DISPLAY
AT(20,20)SIZE(7):HI :: M=M+1
 :: IF M>1 THEN 1050 ELSE 87
0
990 CALL DELSPRITE(#4):: CAL
L SOUND(-100,880,0):: SC=SC+
50 :: DISPLAY AT(22,16)SIZE(
12):SC :: E=E-5 :: DISPLAY A
T(18,18)SIZE(9):E
1000 IF E<5 THEN 1070
1010 HI=MAX(HI,SC):: DISPLAY
 AT(20,20)SIZE(7):HI :: M=M+
1 :: IF M>1 THEN 1050 ELSE 8
70
1020 CALL JOYST(1,X,Y):: CAL
L POSITION(#1,Z1,S1):: IF Z1
<17 AND Y=4 OR Z1>97 AND Y=-
4 THEN B=0
1030 IF S1<34 AND X=-4 OR S1
>223 AND X=4 THEN C=0
1040 CALL MOTION(#1,-Y*B,X*C
):: B,C=2 :: RETURN
1050 CALL DELSPRITE(#3)
1060 GOTO 810
1070 CALL DELSPRITE(ALL):: I
F BO=1 THEN 1110 ELSE SC=SC+
100 :: QU=QU+50 :: DISPLAY A
T(22,16)SIZE(12):SC
1080 CALL SOUND(200,523,0)::
 CALL SOUND(200,659,0)
1090 CALL SOUND(200,784,0)::
 CALL SOUND(200,1047,0)
1100 DISPLAY AT(18,10)SIZE(1
1):"BONUS:100" :: FOR T=1 TO
 500 :: NEXT T
1110 BO=0 :: QU=QU+500 :: IF
 SC>QU THEN 1260
1120 DISPLAY AT(18,10)SIZE(1
1):"DISQUAL."
1130 FOR I=16 TO 2 STEP -1 :
: CALL SCREEN(I):: FOR T=50
TO 1 STEP -10 :: CALL SOUND(
-10,T+1000,0):: NEXT T :: NE
XT I
1140 DATA S,P,I,E,L,E,N,D,E,
,
1150 RESTORE 1140 :: FOR I=7
 TO 25 STEP 2 :: READ A$ ::
IF I=25 THEN 1180
1160 CALL SOUND(50,1000,0)::
 DISPLAY AT(8,I)SIZE(2):A$
1170 NEXT I
1180 FOR I=1 TO 500 :: NEXT
I :: QU=0
1190 DISPLAY AT(18,10)SIZE(1
8):"ENERGIE:";E :: DISPLAY A
T(20,10)SIZE(18):"HIGHSCORE:
";HI :: DISPLAY AT(22,10)SIZ
E(18):"SCORE:";SC
1200 FOR I=1 TO 500 :: NEXT
I
1210 DISPLAY AT(18,10)SIZE(1
7):"WOLLEN SIE NOCH"
1220 DISPLAY AT(20,10)SIZE(1
7):"EINMAL SPIELEN ?" :: DIS
PLAY AT(22,10)SIZE(17):"(Y/N
)"
1230 E=100 :: SC=0 :: Z=0
1240 CALL KEY(0,K,S):: IF (K
=78)+(K=110)THEN CALL CLEAR
:: END
1250 IF (K=89)+(K=121)THEN 5
30 ELSE 1240
1260 CALL DELSPRITE(ALL)
1270 DISPLAY AT(18,10)SIZE(1
7):"FEUERTASTE DRUE-" :: DIS
PLAY AT(20,10)SIZE(17):"CKEN
 WENN DAS [E] " :: DISPLAY A
T(22,10)SIZE(17):"ERSCHEINT"
1280 FOR I=1 TO 500 :: NEXT
I
1290 F=1000
1300 CALL HCHAR(INT(RND*7)+1
7,INT(RND*5)+4,136):: CALL H
CHAR(INT(RND*7)+17,INT(RND*5
)+4,123)
1310 CALL KEY(1,K,S):: IF K=
18 THEN 1390
1320 CALL SOUND(-100,F,0)::
IF F=2000 THEN F=1000 ELSE F
=F+1000
1330 IF RND<.07 THEN 1340 EL
SE 1300
1340 FOR I=17 TO 23 STEP 3 :
: CALL HCHAR(I,4,136,5):: NE
XT I
1350 FOR I=18 TO 19 :: CALL
HCHAR(I,4,136):: CALL HCHAR(
I,5,123,4):: CALL HCHAR(I+3,
4,136,2):: CALL HCHAR(I+3,6,
123,3):: NEXT I
1360 FOR I=1 TO 2 :: CALL SO
UND(-10,1000,0)
1370 CALL KEY(1,K,S):: IF S<
>0 THEN 1410
1380 NEXT I
1390 DISPLAY AT(18,10)SIZE(1
7):"SIE HABEN LEIDER" :: DIS
PLAY AT(20,10)SIZE(17):"KEIN
E FREIENERGIE" :: DISPLAY AT
(22,10)SIZE(17):"BEKOMMEN"
1400 GOTO 1130
1410 DISPLAY AT(18,10)SIZE(1
7):"SIE HABEN WIEDER" :: DIS
PLAY AT(20,10)SIZE(17):"100
ENERGIEEIN-" :: DISPLAY AT(2
```

```
2,10)SIZE(17):"HEITEN BEKOMM
EN"
1420 E=100 :: FOR I=1 TO 500
 :: NEXT I :: GOTO 690
```

BÖRSE

Superangebot!!!!!!
Verkaufe TI 99/4A m P-Box + 3 RS 232 + 32 K-Karte + Disk-Contr. + Laufwerk + P-Code-Karte + EA-Modul + Ext-Basic + Sprachsynth. + Multipl. + TI-Writer + Buchh. + TE II + Schach + Text + Dateiverw. + Statistik + Datenverw. + viel Software + Bücher für sage und schreibe VB 2900,-- . Tel.: 05976/1698.

TI 99/4A + Ex. Bas. + Rec. + Kabel + Literatur + Software (6 CAss + Listing) 450,--, 0207/870101 ab 16.00.

RS 232 Schn.stelle ex. Drukker Brother EP 22, Dateiver. Statistik, Preise VHS. Tel.: 040/6439853.

Verkaufe: Origig. EX-Basic + Handb. 160 DM, Music Maker Mod. 35 DM, Tips und Tricks 15 DM (alles zus. 190 DM). 06162/3922.

Verkaufe: Orig. TI Peri-Box 200,-- DM, Recorder + Kabel 100,-- DM, Ext. Basic mit engl. Handbuch 150,--, Apesoft Exp. Grafic Basic (Disk.) 100,-- DM. Tel.: 06201/69729

TI-Literatur, ältere Magazine aus den USA und d. deuschen Sprachraum zu verkaufen. Liste gegen Rückporto von D. Taube, Dingelstedtwall 16, 3260 Rinteln 1, oder tel: 05751-42913 (ab 16.00).

Billig zu verkaufen: TI 99/4A 3 X-Basic, Joystick + Recorder + Kabel, MBI-Multiboard-Interface auf Centronic (RS 232) + Module: Buchungsjournal u. Statistik + viele Programme, kompl. nur 650,-- DM. Suche: 32 KB Erweiterung extern u. gepuffert. Tel.: 02594/3163 ab 18.00

TI 99/4A 100, Ex-Bas 100 Joyst. 50, Sprachsynth 90, Cassrec. 80, Grafiktab 90, 3 Spielmodu. 30, + Bücher. Tel.: 09721/3940 ab 17 Uhr.

Verkaufe Drucker HR 5 von Brother mit Kabel RS 232C, DM 280,--. (neuwertig). Lucas, 06821/70160.

Verkaufe TI 99/4A + Ext. B. + Cass.-Kabel + Orig.-Joysticks + Prog.-Cassetten 400 DM. R. Möser, Tel: 0541-187471, ab 18.00 Uhr.

HP 7035B X-Y-Schreiber m. Buchsenbel-Planlanonstecker. Sehr gut erhalten. Tel.: 041 74/3245 ab 18 Uhr.

Aufgepasst: Verkaufe: Orig. TI-Diskcontroller mit Diskmanager II für nur 300 DM. S. Wöllner, Darmstadt. Tel.: 06151/311195.

Hallo, TI-User! Ich suche Tauschpartner für Programme in TI und Ex-Basic. Meldet euch bei: Stefan Rossow, Winsener Str. 1, 2358 Kattendorf. Oder ruft ab 19 h an. Tel.: 04191/8353. Tschüß.

Verk. TI-99/4A: Extb. Handb. (TI/Extb.) ca. 35 Spiele in Extb. VB 400 DM. Tel.: 0203/583718 von 20 bis 22 Uhr.

Verkaufe: TI 99/4A + Extended Basic + Schachmodul + Literatur. Preis VB. Dr. R. Schulze, Tel.: 0231/803001.

Verkaufe Minimemory-Modul mit Software! Preis VB 220,-- DM. Suche für TI: Drucker Interface, Drucker und 32 K Speichererw. Angebote an Tel.: 02051/62666 (Velbert).

TI99/4A m. Ext-Basic, Rekorder + Lit. 380,— Module: Pole Position + Jungle Hunt a 50,—, Shamus + Protector a 35,—, TI Invaders; Munchman; Hunt the Wumpus; Jaw breaker; The Attack, Tombstone City, Dhisholm Trail a 25,— Car tridge Expander 3fach 80,—
H. Menzel I.d. Freßäckern 17/1, 7120 Bietigheim-Bissingen (o. Tel. 07142-61549 ab 19 h)

+++ Top Angebot +++
Verk. TI99/4A + Ex. Basic + Sp. Syn. + Joyst. + Parsec + Rec. Kab. (alles in orig. Verp.) + viele Prg. + TI-Intern + Zeitschr., alles 50 % billiger! Preis: nur 475 DM!!! Telefon 0761/71682.

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *    SOLOWUERFELN      *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *     Frajo . Fry      *
17 ! *                      *
18 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     3361 Bytes       *
28 ! *                      *
29 ! ************************
100 DATA 100,70,60,50,40,30,
40,50,60,70,100
110 CALL CLEAR :: CALL SCREE
N(14):: CALL MAGNIFY(3):: DI
M A(12),B(6),C(5),X(2),Y(12)
120 FOR I=2 TO 8 :: CALL COL
OR(I,16,1):: NEXT I :: CALL
CHAR(64,"3C4299A1A199423C")
130 DISPLAY AT(11,9)BEEP:"SO
LO-WUERFELN": : :TAB(9);"@BY
 FRAJO FRY"
140 CALL CHAR(120,"000000000
00003030303"&RPT$("0",24)&"C
0C0C0C",124,"0078787878"&RPT
$("0",44)&"1E1E1E1E")
150 CALL CHAR(128,"000000000
0000303030300787878780000 1E1
E1E1E00C0C0C0C",132,"0078787
878000000000000787878780000 1
E1E1E1E0000000000001E1E1E1E"
)
160 CALL CHAR(136,"007878787
8000303030300787878780000 1E1
E1E1E00C0C0C0C0001E1E1E1E",1
40,"007B7B7B7B0000000000007B
7B7B7B0000DEDEDEDE0000000000
00DEDEDEDE")
170 CALL CHAR(104,"000000FFF
F",105,"1818181818181818000000
00FFFF18181818181FFFF",108,
"1818181F1F18181818181F8F81
81818")
180 CALL CHAR(110,"0000001F1
F1818180000000F8F818181818181
81F1F",113,"181818F8F8",114,
"181818FFFF181818",95,"FFFFF
FFFFFFFFFFFF")
190 RESTORE :: FOR I=2 TO 12
 :: READ Y(I):: A(I)=0 :: IF
 I<8 THEN B(I-1)=0 ELSE C(I-
7)=0
200 NEXT I :: M=17 :: F,P=0
:: CALL CLEAR
210 DISPLAY AT(1,1):"nhhjhjh
```

```
hhhhhjhhhhonhjho...1 i 2i i.
.X100i....ii1i i.....lhhrhrh
hhhhhrhhhhmlhrhm.....i 3i i.
.X 70i....ii2i i...2 lhhrhrh
hhhhhrhhhhmlhrhm"
220 DISPLAY AT(6,1):"i 4i i.
.X 60i....ii3i i.....lhhrhrh
hhhhhrhhhhmlhrhm...3 i 5i i.
.X 50i....ii4i i.....lhhrhrh
hhhhhrhhhhmlhrhm.....i 6i i.
.X 40i....ii5i i...4"
230 DISPLAY AT(11,1):"lhhrhr
hhhhhhrhhhhmlhrhm.....i 7i i
..X 30i....ii6i i.....lhhrhr
hhhhhhrhhhhmphkhq...5 i 8i i
..X 40i....i.........lhhrhr
hhhhhhrhhhhm"
240 DISPLAY AT(16,1):"i 9i i
..X 50i....i.........lhhrhr
hhhhhhrhhhhm.........i10i i
..X 60i....i1.PAAR....lhhrhr
hhhhhhrhhhhm.........i11i i
..X 70i....i2.PAAR"
250 DISPLAY AT(21,1):"lhhrhr
hhhhhhrhhhhm.........i12i i
..X100i....iREST.....phhkhk
hhhhhhkhhhhq.........GESAMT
PUNKTE"
260 FOR I=1 TO 13 STEP 3 ::
FOR J=0 TO 1 :: CALL HCHAR(I
+J,31,95,2):: NEXT J :: NEXT
 I
270 D=0 :: FOR I=1 TO 5 :: R
ANDOMIZE :: C(I)=INT(6*RND)+
1 :: D=D+C(I):: Z=C(I)*4+116
 :: CALL SPRITE(#I,Z,5,I*24-
23,241):: NEXT I
280 N=6 :: FOR I=1 TO 2 :: C
ALL SPALTE(M,N):: FOR J=1 TO
 2
290 CALL KEY(0,K,S):: IF K<4
9 OR K>53 THEN 290 :: CALL G
CHAR((K-48)*3-2,29,U):: IF U
=32 THEN 290
300 CALL HCHAR((K-48)*3-2,29
,32):: CALL HCHAR(16+I*2,26+
J*2,C(K-48)+48):: X(I)=X(I)+
C(K-48)
310 NEXT J :: CALL HCHAR(16+
I*2,29,43):: NEXT I :: N=4 :
: CALL SPALTE(M,N):: CALL VC
HAR(1,29,32,13):: E=D-X(1)-X
(2):: CALL HCHAR(22,28,E+48)
320 CALL GCHAR(E*2,24,U):: I
F U>32 THEN 360 :: F=F+1 ::
IF F<4 THEN 360
330 FOR I=1 TO 5 :: FOR J=1
TO 6 :: IF B(J)=0 THEN 350
340 IF C(I)=J THEN CALL SOUN
D(200,110,0):: CALL LOESCHEN
(X()):: CALL ZEILE :: CALL H
CHAR(23,21,32,4):: GOTO 280
350 NEXT J :: NEXT I :: GOTO
 370
360 B(E)=B(E)+1 :: CALL HCHA
R(E*2,24,B(E)+48)
370 FOR I=1 TO 2 :: A(X(I))=
A(X(I))+1 :: IF A(X(I))>10 T
HEN A(X(I))=10
380 CALL GCHAR(X(I)*2-2,7,U)
:: IF U<53 THEN CALL HCHAR(X
(I)*2-2,7,A(X(I))+48)ELSE CA
LL HCHAR(X(I)*2-2,10,A(X(I))
+43)
390 NEXT I :: CALL DELSPRITE
(ALL):: CALL LOESCHEN(X()):
 CALL ZEILE :: IF B(E)<8 THE
N 270
400 FOR I=2 TO 12 :: IF A(I)
=0 THEN A(I)=A(I)+5
410 IF A(I)>4 THEN Q=(A(I)-5
)*Y(I):: DISPLAY AT(I*2-2,17
-LEN(STR$(Q)))SIZE(LEN(STR$(
Q))+1):Q :: GOTO 430
420 Q=-200 :: DISPLAY AT(I*2
-2,14)SIZE(4):Q
430 P=P+Q :: NEXT I :: DISPL
AY AT(24,14):P
440 CALL KEY(0,K,S):: IF K=3
2 THEN 190 ELSE IF K<>13 THE
N 440 :: END
450 SUB LOESCHEN(X()):: FOR
I=1 TO 750 :: NEXT I :: FOR
I=18 TO 22 STEP 2 :: CALL HC
HAR(I,28,32,3):: NEXT I :: X
(1),X(2)=0 :: SUBEND
460 SUB ZEILE :: FOR I=1 TO
5 :: CALL HCHAR(I*3-2,29,I+4
8):: NEXT I :: SUBEND
470 SUB SPALTE(M,N):: M=M+2
:: DISPLAY AT(M-2,19)SIZE(N+
2):: IF M=25 THEN M=19
480 DISPLAY AT(M,19)SIZE(N)B
EEP:"--------" :: SUBEND
```

LISTINGS

LONG LIVE THE KING

Dieses Programm ist auch ein gutes Training für Schachanfänger, um die Angriffe von Figuren abschätzen zu lernen und die eigene Figur (in diesem Fall den König) optimal positionieren zu lernen, sowohl defensiv, als auch offensiv.

Nutzen Sie unseren kostenlosen Kleinanzeigenservice

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! * LONG LIVE THE KING   *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *     Frajo  Fry       *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     4838 Bytes       *
28 ! *                      *
29 ! ************************
100 DATA 1,6,80,41,0,10,112,
41,-1,14,144,41,-2,18,176,41
,1,6,80,73,0,10,112,73,-1,14
,144,73,-2,18
110 DATA 176,73,1,8,96,105,0
,12,128,105,-1,16,160,105,2,
6,80,137,1,10,112,137,0,14,1
44,137,-1,18,176,137
120 DATA 83,67,79,82,69,48,4
9,50,51,52,53,54,55,56,57
130 DATA 108,108,108,108,112
,120,108,112,120,128,108,112
,120,128,108,112,120,128,136
,108,112,120,128,136
140 CALL CLEAR :: DIM E$(2),
Z$(14),F(7),G(7):: CALL CHAR
(64,"3C4299A1A199423C")
150 FOR I=1 TO 15 :: CALL SP
RITE(#I,ASC(SEG$("LONGLIVETH
EKING",I,1)),5,I*8+25,33)::
NEXT I :: FOR I=1 TO 15 :: R
EAD A,B,C,D :: CALL MOTION(#
I,A,B)
160 CALL POSITION(#I,X,Y)::
IF Y<C THEN 160 :: CALL MOTI
ON(#I,0,0):: CALL LOCATE(#I,
D,C+1)
170 NEXT I :: DISPLAY AT(23,
9)BEEP:"@ BY FRAJO FRY" :: F
OR I=1 TO 12 :: CALL MAGNIFY
(1):: CALL SOUND(150,440,0):
: CALL MAGNIFY(2):: CALL SOU
ND(150,I*110,0):: NEXT I
180 CALL CHAR(96,"FF80808080
808080FF",98,"FF0101010101
0180808080808080800",101,"01
0101010101010180808080808080
FF0000000000000FF0101010101
0101FF")
190 CALL CHAR(108,"000000000
1030204040402020408080F00000
00080C0402020204040201010F",
112,"0000000304080908080808080
808080F0F000090E0205008C4B49
840201010F0F")
200 CALL CHAR(120,"000001030
405090B0B0909040F041F3F00008
0C020A090D0D0909020F020F8FC"
,128,"0000191F180F0808080808
080F183F3F000098F818F0101010
101010F018FCFC")
210 CALL CHAR(136,"21524C242
2101008080080F0808183F3F844A3
22444080810101 0F0101018FCFC"
,140,"01031D23414141212 11F08
0F0F081F3F80C0B8C48282828484
F810F0F010F8FC")
220 A$=RPT$("`ab",8):: B$=RP
T$("cde",8):: C$=RPT$("fgh",
8):: D$=RPT$("0",48)
230 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL MAGNIFY(3):
: CALL COLOR(9,2,16,10,2,16,
11,14,16,12,5,16,13,13,16,14
,11,16)
240 RESTORE 120 :: FOR I=9 T
O 23 :: READ A :: CALL CHARP
AT(A,Z$(I-9)):: NEXT I :: FO
R I=9 TO 23 :: CALL CHAR(I*4
,Z$(I-9)&D$):: NEXT I
250 FOR I=1 TO 8 :: PRINT A$
,B$,C$;:: NEXT I :: X=9 :: Y
=89 :: CALL SPRITE(#1,140,7,
X,Y):: P=0
260 FOR I=9 TO 16 :: CALL SP
RITE(#I,I*4+(I>14)*4+(I=16)*
4,5,I*16-115-(I>13)*16,225):
: NEXT I
270 DISPLAY AT(1,27):"!" ::
CALL KEY(0,K,S):: IF K<49 OR
 K>57 THEN 270 :: ON K-48 GO
TO 280,290,300,310,360,320,3
30,340,350
280 X=X-24-192*(X=9):: Y=Y-2
4-24*(Y=17):: GOTO 360
290 X=X-24-192*(X=9):: GOTO
360
300 X=X-24-192*(X=9):: Y=Y+2
4+24*(Y=185):: GOTO 360
310 Y=Y-24-24*(Y=17):: GOTO
360
320 Y=Y+24+24*(Y=185):: GOTO
 360
330 X=X+24+192*(X=177):: Y=Y
-24-24*(Y=17):: GOTO 360
340 X=X+24+192*(X=177):: GOT
O 360
350 X=X+24+192*(X=177):: Y=Y
+24+24*(Y=185)
360 M=(X+7)/8 :: N=(Y+7)/8 :
: CALL GCHAR(M,N,Z):: IF Z>1
07 THEN CALL L(M,N):: CALL S
OUND(100,-4,0):: P=P-(Z=108)
-(Z=112)*3-(Z=120)*3-(Z=128)
*4-(Z=136)*8
370 DISPLAY AT(1,27):: CALL
```

```
LOCATE(#1,X,Y):: F(0),F(1)=M
-3 :: F(2),F(3)=M-6 :: F(4),
F(5)=M+3 :: F(6),F(7)=M+6
380 G(0),G(4)=N-6 :: G(1),G(
5)=N+6 :: G(2),G(6)=N-3 :: G
(3),G(7)=N+3 :: FOR I=0 TO 7
 :: IF F(I)<2 OR F(I)>23 OR
G(I)<3 OR G(I)>24 THEN 400
390 CALL GCHAR(F(I),G(I),Z):
: IF Z=112 THEN 630
400 NEXT I :: FOR I=N-3 TO 3
 STEP -3 :: CALL GCHAR(M,I,Z
):: IF Z>107 THEN IF Z<128 T
HEN 420 ELSE 630
410 NEXT I
420 FOR I=N+3 TO 24 STEP 3 :
: CALL GCHAR(M,I,Z):: IF Z>1
07 THEN IF Z<128 THEN 440 EL
SE 630
430 NEXT I
440 FOR I=M-3 TO 2 STEP -3 :
: CALL GCHAR(I,N,Z):: IF Z>1
07 THEN IF Z<128 THEN 460 EL
SE 630
450 NEXT I
460 FOR I=M+3 TO 24 STEP 3 :
: CALL GCHAR(I,N,Z):: IF Z>1
07 THEN IF Z<128 THEN 480 EL
SE 630
470 NEXT I
480 W=M :: FOR I=N+3 TO N+M-
2+(M+N>26)*(M+N-26)STEP 3 ::
 W=W-3 :: CALL GCHAR(W,I,Z):
: IF Z>107 THEN IF (Z=120)+(
Z=136)=0 THEN 500 ELSE 630
490 NEXT I
500 W=M :: FOR I=N-3 TO N-M+
2-(M-N>1)*(M-N+1)STEP -3 ::
W=W-3 :: CALL GCHAR(W,I,Z)::
 IF Z>107 THEN IF (Z=120)+(Z
=136)=0 THEN 520 ELSE 630
510 NEXT I
520 W=M :: FOR I=N+3 TO N-M+
23+(N-M>1)*(N-M-1)STEP 3 ::
W=W+3 :: CALL GCHAR(W,I,Z)::
 IF I=N+3 THEN IF Z=108 THEN
 630
530 IF Z>107 THEN IF (Z=120)
+(Z=136)=0 THEN 550 ELSE 630
540 NEXT I
550 W=M :: FOR I=N-3 TO N+M-
23-(M+N<26)*(26-M-N)STEP -3
:: W=W+3 :: CALL GCHAR(W,I,Z
):: IF I=N-3 THEN IF Z=108 T
HEN 630
560 IF Z>107 THEN IF (Z=120)
+(Z=136)=0 THEN 580 ELSE 630
570 NEXT I
580 M=M+3+24*(M=23):: CALL G
CHAR(M,N,Z):: IF Z>107 THEN
CALL L(M,N)
```

LISTINGS

```
590 PRINT A$,B$,C$;:: A=MIN(
(P+50)/50,3):: RANDOMIZE ::
FOR I=1 TO INT(RND*A+.1):: B
=MIN(P/6,22):: RESTORE 130 :
: FOR J=1 TO INT(RND*B+A)::
READ C :: NEXT J
600 Q=INT(RND*8+1)*3 :: CALL
 GCHAR(23,Q,Z):: IF Z>104 OR
(X=177 AND N=Q)THEN 600 :: C
ALL HCHAR(23,Q,C):: CALL HCH
AR(23,Q+1,C+2):: CALL HCHAR(
24,Q,C+1)
610 CALL HCHAR(24,Q+1,C+3)::
 NEXT I :: P=P+1 :: P1=INT(P
/100):: P2=INT((P-P1*100)/10
):: P3=P-P1*100-P2*10
620 CALL PATTERN(#14,56+P1*4
,#15,56+P2*4,#16,56+P3*4)::
GOTO 270
630 FOR I=9 TO 16 :: CALL DE
LSPRITE(#I):: NEXT I :: FOR
I=14 TO 2 STEP -1 :: CALL CO
LOR(#1,I):: CALL SOUND(200,1
00+I*30,0,50+I*30,0,-8,0)::
NEXT I
640 CALL CHARSET :: H=MAX(P,
H):: E$(0)="SCORE.."&STR$(P)
:: E$(1)="HIGHSCORE.."&STR$(
H):: E$(2)="NEW GAME...Y OR
N?"
650 FOR I=0 TO 2 :: FOR J=1
TO LEN(E$(I)):: CALL VCHAR(7
-I*2+J,28+I*2,ASC(SEG$(E$(I)
,J,1)))):: NEXT J :: NEXT I
660 CALL KEY(0,K,S):: IF K=8
9 THEN 230 ELSE IF K<>78 THE
N 660 :: CALL CLEAR :: END
670 SUB L(M,N):: CALL HCHAR(
M,N,99):: CALL HCHAR(M,N+1,1
00):: CALL HCHAR(M+1,N,102):
: CALL HCHAR(M+1,N+1,103)::
SUBEND
```

POGO - JUMP

Der Held dieses Spiels ist Pogo, ein fanatischer Vegetarier, dessen einzige Lebensaufgabe es ist, Früchte zu sammeln. Dabei lauern natürlich vielfältige Gefahren, die geschickt übersprungen werden müssen: Ameisenhaufen bilden gefährliche Hindernisse und Abgründe gefährden sein Leben. Achtung, Pogo wechselt automatisch von Etage zu Etage!!!
Sind endlich alle Früchte aufgegessen, ist er so gestärkt, daß ihm Flügel wachsen und er ein Bonusleben bekommt. Nun beginnt die Bonusrunde: Am unteren Bildschirmrand erscheinen drei Schlangen. Pogo fliegt über sie hinweg und wartet auf den Befehl des Spielers, der die Aufgabe hat, im richtigen Moment den Feuerknopf zu drükken, worauf Pogo langsam zu Boden schwebt, wo jedoch schon die Schlangen auf ihn warten. Schafft Pogo es, an

Bitte lesen Sie weiter auf Seite 141

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *      POGO-JUMP       *
13 ! *                      *
14 ! *     Copyright by     *
15 ! *                      *
16 ! *   Dirk  Theissen     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *     Joystick 1       *
23 ! * (opt. Disk-Laufw.)   *
24 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    10180 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CHAR(128,"FF55FF996
6996681",100,"01070F1F3F3F7F
7F80E0F0F8FCFCFEFE")
110 ON ERROR 120 :: CALL CLE
AR :: RANDOMIZE :: P=1 :: DI
M F1$(16),F2$(16):: OPEN #1:
"DSK1.HI-SCORE" :: INPUT #1:
HI :: CLOSE #1
120 ON ERROR 1280 :: C$="030
F0404070402010202020201030
0E0F89090F01020C0A0A0A0A0C
0E080" :: CALL CHAR(96,C$)
130 FOR I=1 TO 8 :: FOR J=1
TO I :: F1$(I)=F1$(I)&SEG$(C
$,J*2-1,2):: F2$(I)=F2$(I)&S
EG$(C$,31+J*2,2):: NEXT J ::
 NEXT I
140 FOR I=1 TO 8 :: FOR J=1
TO I :: F1$(I+8)=F1$(I+8)&SE
G$(C$,J*2+15,2):: F2$(I+8)=F
2$(I+8)&SEG$(C$,47+J*2,2)::
NEXT J :: NEXT I
150 CALL SCREEN(2):: CALL CL
EAR :: CALL MAGNIFY(3):: FOR
 I=1 TO 7 :: CALL COLOR(I,IN
T(RND*13+3),1):: NEXT I
160 CALL COLOR(12,9,1):: CAL
L ZEICHENSATZ
170 B$=RPT$("F",16):: CALL C
HAR(95,B$,113,B$,131,B$):: D
ISPLAY AT(1,1):"Cq_Cq_Cq_Cq_
Cq_Cq_Cq_Cq_Cq_C" :: B$="_qC
_qC_qC_qC_qC_qC_qC_q"
180 FOR I=2 TO 24 :: CALL HC
HAR(I,3,ASC(SEG$(B$,I-1,1)))
:: NEXT I :: DISPLAY AT(24,1
):"qC_qC_qC_qC_qC_qC_qC_qC_q
C_q"
190 FOR I=24 TO 2 STEP -1 ::
 CALL HCHAR(I,30,ASC(SEG$(B$
,25-I,1))):: NEXT I :: CALL
SPRITE(#1,96,15,144,100):: D
ISPLAY AT(21,11):",,,";
200 DISPLAY AT(22,3):"(C)198
5.....DIRK THEISEN";:: DISPL
AY AT(9,7):"SAMMELN SIE ALLE
";:: DISPLAY AT(11,3):"FRUEC
HTE,AEPFEL USW,EIN.";
210 DISPLAY AT(14,3):" STEUE
RUNG:...JOYSTICK";:: DISPLAY
 AT(16,3):" JUMP:TRIGGER+RIC
HTUNG";:: DISPLAY AT(18,19):
"HI:";:: DISPLAY AT(18,22):U
SING "####":HI;
220 DISPLAY AT(5,10):"q_Cq_C
q_Cq_";:: DISPLAY AT(6,10):"
CPOGO-JUMPC";:: DISPLAY AT(7
,10):"_qC_qC_qC_q";
230 CALL COLOR(8,4,1,11,9,1,
13,12,1):: CALL J(P):: CALL
COLOR(8,9,1,11,12,1,13,4,1):
: CALL J(P)
240 CALL COLOR(9,12,1,11,4,1
,13,9,1):: CALL J(P):: J=J+1
 :: IF J<40 THEN 230 ELSE CA
LL DELSPRITE(ALL):: CALL CLE
AR
250 CALL CHAR(132,"030F04040
70442717F3E3E3E020103O0E0F89
090F01021C7FFBEBEBCA0C0E080"
)
260 CALL COLOR(13,11,2,9,14,
1):: LV=0 :: L=2
270 CALL CHAR(108,"000000000
000000000000C0E010190F0600000
000000000000000000060F0990F06"
)
280 CALL CHAR(130,"81FF81818
1FF8181",104,RPT$("0",64),12
0,"60107CFEFEFE7C")
290 CALL CHAR(104,"000000000
000000000000000060F99F06000000
000000000000000003070898F060"
)
300 GOTO 610
310 IF LV>4 THEN GOSUB 1190
:: GOTO 330
320 CALL DELSPRITE(ALL):: CA
LL CLEAR :: ON LV GOSUB 980,
1080,980,1190 :: DISPLAY AT(
1,8):"LEVEL:"&NAM$
330 FOR I=L TO 1 STEP -1 ::
DISPLAY AT(1,I*2+21):"`b";::
 NEXT I :: CALL GAMI :: DISP
LAY AT(1,1):"\]x^[";
340 WRT$="HIGH SCORE "&STR$(
HI):: FOR I=1 TO LEN(WRT$)::
 CALL HCHAR(I+4,32,ASC(SEG$(
WRT$,I,1))):: NEXT I
350 IF LV=1 THEN 380
360 CALL SPRITE(#2,108,13,16
,INT(RND*254+1),0,-LV*3,#3,1
04,13,48,INT(RND*254+1),0,LV
*3,#4,108,13,80,INT(RND*254+
```

```
1),0,-1-LV*3)
370 CALL SPRITE(#5,104,13,11
2,INT(RND*254+1),0,1+LV*3)
380 CALL SPRITE(#1,96,2,144,
172):: CALL POGO(F1$(),F2$()
)
390 CALL HCHAR(24,1,32,32)
400 FOR I=1 TO 200 :: NEXT I
 :: P2=172 :: P1=144 :: P=1
:: E=5 :: CALL SOUND(1500,15
00,0):: FOR I=1 TO 250 :: NE
XT I
410 P=-P :: CALL LOCATE(#1,P
1+P,P2):: CALL SOUND(4,-(6+(
P/2-1)),0):: T=T+1
420 ON VAL(SEG$(A$(E),P2/8-.
5,1))GOTO 430,640,520,640,58
0
430 CALL JOYST(1,X,Y):: P2=P
2+X*2 :: CALL KEY(1,K,S)
440 CALL COINC(ALL,CO):: IF
CO THEN 640
450 IF S<>0 THEN 470
460 GOTO 410
470 IF X=0 THEN 410
480 CALL LOCATE(#1,P1-7,P2)
490 FOR I=1 TO 3 :: CALL SOU
ND(-10,I*10+700,0):: P2=P2+X
 :: CALL LOCATE(#1,P1-7,P2):
: IF X=0 THEN K1=5 ELSE K1=0
500 FOR K1=1 TO K1 :: NEXT K
1 :: NEXT I
510 CALL LOCATE(#1,P1,P2+X):
: P2=P2+X :: GOTO 420
520 IF T>800 THEN DISPLAY AT
(24,8):"OUT OF TIME" :: T=10
0 :: L=0 :: GOTO 640
530 P3=P1/8+3 :: P4=P2/8-.5
540 CALL GCHAR(P3+1,P4+2,GH)
:: IF GH=130 THEN ST=4 ELSE
ST=-4
550 J=P1+32*(ST/4):: FOR I=P
1 TO J STEP ST :: CALL LOCAT
E(#1,I,P2):: CALL SOUND(-100
,223+(I-P1)*-2,1):: NEXT I :
: P1=P1+32*(ST/4):: E=E+ST/4
560 P3=P1/8+3 :: P4=P2/8-.5
570 GOTO 430
580 CALL SOUND(30,650,0):: C
ALL HCHAR(E*4,P2/8+1.5,32)::
 A$(E)=SEG$(A$(E),1,P2/8-1.5
)&"1"&SEG$(A$(E),P2/8+.5,30-
(P2/8-1.5))
590 SC=SC+1 :: CALL HCHAR(22
,SC*2+1-20*(LV-1),120):: IF
SC<10*LV THEN 410
600 GOTO 830
610 BO=BO+10*LV :: CALL DELS
PRITE(ALL):: FOR I=1 TO 5 ::
 A$(I)="" :: NEXT I :: CALL
CLEAR :: DISPLAY AT(8,6):"YO
U ENTER LEVEL";LV+1;"." :: D
ISPLAY AT(10,7):"TOTAL SCORE
:";LV*100+BO
620 FOR I=1 TO 40 :: CALL SO
UND(-5,I*3+210,3,I*3+211,5,I
*3+209,5,-6,14):: NEXT I
630 LV=LV+1 :: L=L+1+(L=3)::
 GOTO 310
640 T=T-15 :: CALL POSITION(
#1,P1,P2):: CALL SOUND(3000,
-6,10):: CALL SPRITE(#20,96,
13,P1,P2)
650 CALL MOTION(#1,-10,1)::
FOR I=1 TO P1/5 :: CALL PATT
ERN(#1,132):: FOR J=1 TO 5 :
: NEXT J :: CALL PATTERN(#1,
96)
660 FOR J=1 TO 5 :: NEXT J :
: NEXT I
670 CALL COLOR(#1,2):: CALL
MOTION(#1,0,0):: CALL LOCATE
(#1,145,172):: CALL DELSPRIT
E(#20)
680 FOR I=1 TO 3 :: CALL SOU
ND(500,210-I*30,0,211-I*30,3
,209-I*30,3):: NEXT I :: CAL
L HCHAR(1,31-L*2,32,2):: L=L
-1 :: IF L=0 THEN 740
690 FOR I=1 TO 100 :: NEXT I
700 CALL POGO(F1$(),F2$())
710 CALL CHAR(96,C$)
720 CALL HCHAR(24,1,32,32)
730 FOR I=1 TO 15 :: NEXT I
:: GOTO 400
740 DISPLAY AT(11,11):"GAME
OVER";
750 FOR I=1 TO 40 :: DISPLAY
 AT(11,11):".........";:: DI
SPLAY AT(11,11):"GAME OVER";
:: NEXT I
760 IF SC*10+BO>HI THEN HI=S
C*10+BO
770 DISPLAY AT(24,1):"PL-SCO
RE:";STR$(SC*10+BO);"..HI-SC
ORE:";STR$(HI):: CALL VCHAR(
1,32,32,22)
780 IF HI=SC*10+BO THEN ON E
RROR 790 :: CALL TWIT(2):: O
PEN #1:"DSK1.HI-SCORE" :: PR
INT #1:HI :: CLOSE #1
790 ON ERROR 1280 :: BO=0 ::
 FOR I=111 TO 140 STEP 3 ::
CALL SOUND(-100,I,5,I-1,18,I
+1,18)
800 CALL KEY(1,K,S):: IF S T
HEN LV=0 :: L=3 :: SC=0 :: G
OTO 260
810 IF K=ASC("E")OR K=ASC("e
")THEN CALL CLEAR :: PRINT "
(C) 1984 DIRK THEISEN": : ::
 END
```

LISTINGS

LISTINGS

```
820 NEXT I :: SC=0 :: LV=0 :
: L=2 :: GOTO 790
830 CALL DELSPRITE(#1,#2,#3,
#4,#5):: FOR I=500 TO 602 ST
EP 2 :: CALL SOUND(-600,I,0)
:: NEXT I
840 CALL LOCATE(#1,200,9)
850 IF T<80+LV*20+(LV>1)*-27
 THEN CALL CLEAR :: DISPLAY
AT(6,4):"TIME BONUS:";STR$(L
V*6+(190-T)):: CALL TWIT(2):
: BO=BO+LV*6+(190-T):: T=0
860 CALL CLEAR :: DISPLAY AT
(1,9):"BONUS-ROUND" :: FOR I
=1 TO 3 :: CALL SPRITE(#(I+1
),108,16,175,I*(255/3),0,-(3
+4*LV)):: NEXT I :: CALL TWI
T(3)
870 CALL LOCATE(#1,10,127)::
 CALL MOTION(#1,LV*2+1,10+LV
*3)
880 CALL PATTERN(#1,132):: C
ALL POSITION(#1,P,P2):: IF P
>150 THEN 910
890 CALL PATTERN(#1,96)
900 CALL KEY(1,K,S):: IF NOT
 S THEN 880
910 CALL MOTION(#1,9,0):: CA
LL PATTERN(#1,132)
920 CALL SOUND(-300,400-P1,8
):: CALL COINC(ALL,C):: IF C
 THEN 960 ELSE CALL POSITION
(#1,P1,P2):: IF P1>200 THEN
930 ELSE 920
930 P=INT((162-P)/2):: CALL
MOTION(#1,0,0):: DISPLAY AT(
21,1):" BONUS..BONUS..BONUS.
.BONUS...";P;"...";P;"...";P
;"...";P
940 FOR I=1 TO 4 :: FOR J=40
0 TO 350 STEP -3 :: CALL SOU
ND(-100,J,3,J+1,5,J-1,5):: N
EXT J :: NEXT I :: BO=BO+P
950 CALL CLEAR :: GOTO 610
960 CALL MOTION(#1,0,0):: DI
SPLAY AT(20,1):" FAILED ! FA
ILED ! FAILED !": :"...BONUS
: 000" :: CALL LOCATE(#1,200
,P2)
970 FOR I=1 TO 4 :: FOR J=18
0 TO 110 STEP -3 :: CALL SOU
ND(-200,J,4,-7,5):: NEXT J :
: NEXT I :: CALL CLEAR :: GO
TO 610
980 DISPLAY AT(5,1):",,,--,,
,,,,,,,,,--,,,,,,,,,,," :: DI
SPLAY AT(9,1):",,,,,,,,,-,,,
--,--,,,,--,,,,"
990 DISPLAY AT(13,1):",,,,--
,,,,,,,,,---,,,,--,,,," :: D
ISPLAY AT(16,10):",,,,,,,--,
,,,--,,,-,,,,--,,"
1000 A$(1)="1312211144111512
22131114411522" :: A$(2)="13
11445122131225221312211312222
" :: A$(3)="1131221544131122
22513122134422"
1010 A$(4)="5134413221441221
12213122152222" :: A$(5)="15
441131111111144111111111522
"
1020 CALL HCHAR(21,3,128,28)
:: NAM$="APPLES" :: IF LV=3
THEN NAM$="ORANGES" :: CALL
CHAR(120,"387CFEFEFE7C38")
1030 RESTORE 980 :: FOR I=2
TO 8 :: READ A :: CALL VCHAR
(INT(I/2)*4,A,130,5):: NEXT
I
1040 FOR I=2 TO 11 :: READ A
 :: CALL HCHAR(INT(I/2)*4,A,
100):: CALL HCHAR(INT(I/2)*4
,A+1,101):: NEXT I
1050 FOR I=2 TO 11 :: READ A
 :: CALL HCHAR(INT(I/2)*4,A,
120):: NEXT I
1060 DATA 4,22,14,28,5,23,9,
11,26,7,7,11,29,6,13,5,19,16
,30,9,18,10,21,3,28,4,30
1070 RETURN
1080 DISPLAY AT(5,1):",,,,,,
,,,,,--,,,,,,,,,,,,,,," :: D
ISPLAY AT(9,1):",,,,,,,,,,,,
,---,,,,,,,,,,,," :: DISPLAY
 AT(13,1):",,,,,,,,,,,,,,,--
----,,,,-,,"
1090 DISPLAY AT(17,1):",,,,,
,,---,,,,,,,,,,,,,,,,,,," ::
CALL HCHAR(21,3,128,28):: NA
M$="CHERRIES"
1100 A$(1)="1314411511122211
1344111511122" :: A$(2)="53
111441111312221314411135112
2" :: A$(3)="1131441154431512
22322111321122"
1110 A$(4)="1131513222344111
11314411151122" :: A$(5)="51
441131513114411111111111122
"
1120 CALL CHAR(130,"18041820
18041820")
1130 RESTORE 1080 :: FOR I=2
 TO 9 :: READ A :: CALL VCHA
R(INT(I/2)*4,A,130,5):: NEXT
 I
1140 CALL CHAR(120,"10182442
E6F7F762")
1150 FOR I=2 TO 11 :: READ A
 :: CALL HCHAR(INT(I/2)*4,A,
120):: NEXT I
1160 FOR I=2 TO 11 :: READ A
 :: CALL HCHAR(INT(I/2)*4,A,
```

```
100):: CALL HCHAR(INT(I/2)*4
,A+1,101):: NEXT I
1170 DATA 4,20,14,27,5,21,9,
13,10,26,3,28,11,16,7,28,3,1
1,6,21,8,22,7,12,14,23,5,16
1180 RETURN
1190 DISPLAY AT(5,1):",,-,,
,,,,,,,,,,,,,,,,,,,,,,," :: D
ISPLAY AT(9,1):",,,,,,,,,,,,
,,,,,,,,,,,,,,," :: DISPLAY
 AT(13,1):",,,,,,,,,,,,,,,,,
,,,,,,,,,,,"
1200 DISPLAY AT(17,1):",,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,," ::
DISPLAY AT(21,1):",,,,,,,,,,
,,,,,,,,,,,,,,,,,"
1210 A$(1)="31221114411111111
11112544115122" :: A$(2)="32
31134412512211212114412215222
" :: A$(3)="5134434411122121
12225112132122"
1220 A$(4)="51131221521441311
12112144333122" :: A$(5)="44
13214411121251111111122253122
"
1230 CALL COLOR(12,11,1):: R
ESTORE 1190 :: FOR I=2 TO 9
:: READ A :: CALL VCHAR(INT(
I/2)*4,A+2,130,5):: NEXT I :
: CALL CHAR(120,"00000081C37
E3C")
1240 FOR I=2 TO 11 :: READ A
 :: CALL HCHAR(INT(I/2)*4,A+
2,100):: CALL HCHAR(INT(I/2)
*4,A+3,101):: NEXT I
1250 FOR I=2 TO 11 :: READ A
 :: CALL HCHAR(INT(I/2)*4,A+
2,120):: NEXT I :: NAM$="BAN
ANAS"
1260 DATA 1,1,3,6,15,26,4,27
,8,23,7,22,4,7,12,24,1,7,22,
27,11,28,1,21,1,9,15,26
1270 RETURN
1280 ON ERROR 1280 :: GOTO 6
40
1290 SUB GAMI
1300 CALL CHAR(92,"0044AA8A8
AEEAA4A00A4E0A4A4A0A4A400E48
A8AEA2A2AE4",91,"00EE8484C48
48484")
1310 SUBEND
1320 SUB ZEICHENSATZ
1330 RESTORE 1350
1340 FOR I=65 TO 90 :: READ
A$ :: CALL CHAR(I,A$):: NEXT
 I
1350 DATA 007C8282FEC2C2C2,0
0FC8282FCC2C2FC,007C8280C0C0
C27C,00FC8282C2C2C2FC,00FE80
80F8C0C0FE,00FE8080F8C0C0C0,
007E8080CEC2C27C
```

LISTINGS

```
1360 DATA 00828282FEC2C2C2,0
0101010101818,00FE02020606
867C,00888888FCC4C4C4,008080
80C0C0C0FE
1370 DATA 0082C6AA92C2C2C2,0
082C2A292CAC6C2,00FE8282C2C2
C2FE,00FC8282FCC0C0C0,00FE82
82C2CAC6FE,00FC8484FEC2C2C2
1380 DATA 00FE8080FE0606FE,0
0FE101018181818,00828282C2C2
C2FE,0082824464383810,00C2C2
C292AAC682,008282827CC2C2C2,
0082824428303030
1390 DATA 00FE040838204EFE
1400 FOR I=48 TO 57 :: READ
A$ :: CALL CHAR(I,A$):: NEXT
 I
1410 DATA 007C868A92A2C27C,0
0101010101010,00FC02027C80
807E,00FC02021C0202FC,008082
82FE020202,00FE80807C0202FC
1420 DATA 007E8080FC82827C,0
07C040404040404,007C82827C82
827C,007C82827C0202FC
1430 SUBEND
1440 SUB J(P):: P=-P :: CALL
 LOCATE(#1,144+P,100):: CALL
 SOUND(-6,-(6+(P/2-1)),4)::
SUBEND
1450 SUB TWIT(X):: FOR TW=1
TO X :: FOR IT=300 TO 400 ST
EP 5 :: CALL SOUND(-100,IT,2
,-5,10):: NEXT IT :: NEXT TW
 :: SUBEND
1460 SUB POGO(F1$(),F2$()):
 CALL CHAR(96,RPT$("0",64)):
: CALL COLOR(#1,10):: FOR I=
1 TO 8 :: CALL CHAR(96,F1$(I
),98,F2$(I)):: CALL SOUND(-7
0,300+I*3,0):: NEXT I
1470 FOR I=1 TO 8 :: CALL CH
AR(97,F1$(I+8),99,F2$(I+8)):
: CALL SOUND(-70,324+I*3,0):
: NEXT I :: SUBEND
```

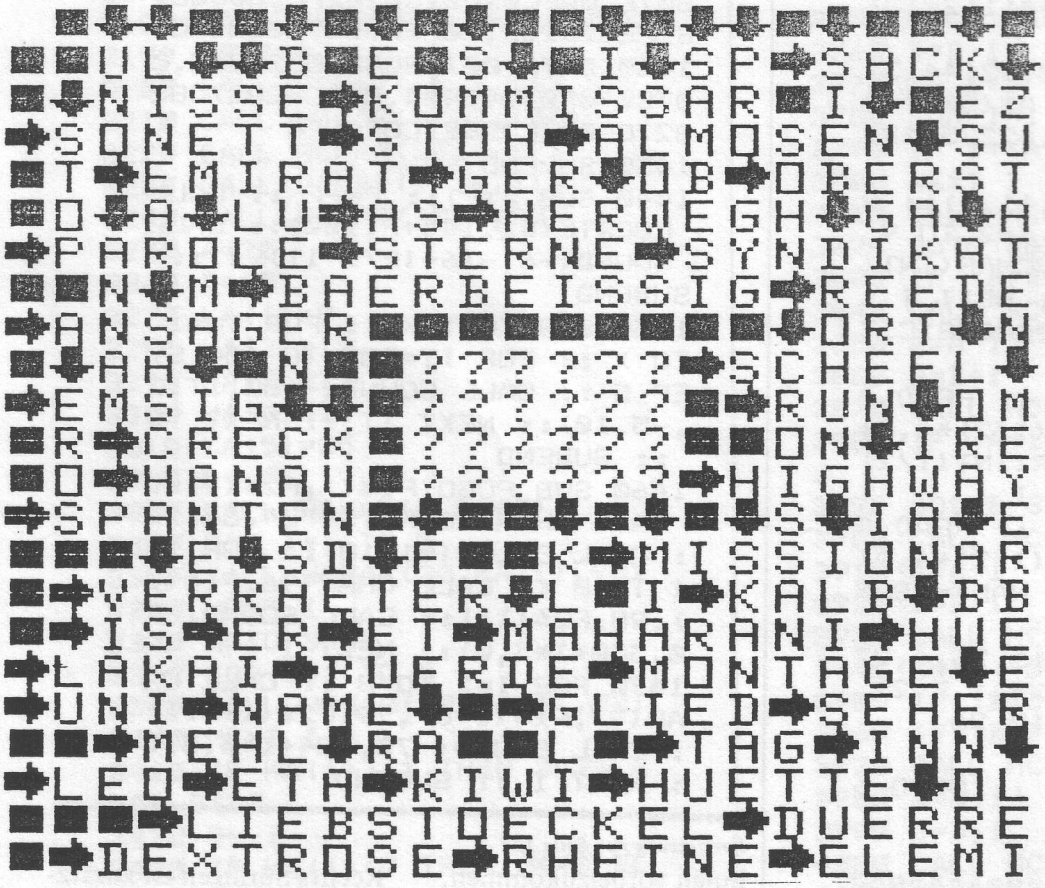
Fortsetzung von Seite 137

ihnen vorbeizukommen, so erhält der Spieler einen Punktebonus. Es beginnt der nächste der insgesamt vier Level, bei denen nun jedoch noch Giftschlangen hinzukommen, die von Runde zu Runde schneller werden. Für geübte Pogo-Jump-Spieler: Wird eine Runde in einer besonders guten Zeit bewältigt, so winkt ein Zeitbonus. Läßt man sich zu viel Zeit, heißt es 'out of time'.
Die Steuerung erfolgt mit Joystick(1). Zum Links/Rechts-Springen ist zusätzlich zur entsprechenden Bewegung des Steuerknüppels der Feuerknopf zu drücken.
Das Spiel verfügt über eine erfreuliche, ungewöhnlich hohe Geschwindigkeit und die Grafik ist fast professionell zu nennen!

Wichtige Hinweise:
1. Vor dem Laden des Programms eventuell CALL FILES (1) und NEW eingeben.
2. Wer nicht über ein Disk-Laufwerk verfügt, muß die

Zeilen 110 und 780 (Hi-Score-Speicherung) für Cassette abändern oder die Dateibefehle weglassen.
3. In den folgenden Zeilen sind in den DISPLAY-Befehlen und der Definition von B$ der Großbuchstabe 'C' und die Kommata ',' zusammen mit der CTRL-Taste einzugeben: 170, 180, 190, 220, 980, 1080, 1090, 1200. Sonst stimmt die Grafik nicht!

KREUZWORT-RÄTSEL



Das Kreuzworträtsel

Dieses Programm wird die Freunde von Kreuzworträtseln nicht nur zur Lösung anregen, sondern auch zur Erstellung neuer, da der Aufbau leicht ersichtlich ist. Bei der Eingabe des Programms ist auf Groß- und Kleinbuchstaben genau zu achten. Nach dem Starten muß die Alpha-Lock-Taste ausgerastet werden. Danach ist das Speichermedium (Cassette oder Disk) einzugeben, auf dem das Kreuzworträtsel eventuell zwischengespeichert werden soll. Im Falle eines Disk-Laufwerks wird eine Datei mit dem Namen 'KREUZ' erstellt. Zur Bedienung: Der blinkende Cursor ist mit dem Joystick 1 auf einen der Pfeile zu bringen und der Feuerknopf zu drücken. Nun erscheint in Laufschrift die bige Frage 'SPEICHERN' ein. Beim nächsten Start des Programms kann der alte Spielstand dann wieder geladen werden, wenn man die Frage 'gespeichertes Rätsel einlesen' mit der J-Taste beantwortet.

Frage. Nach neuerlichem Drücken des Feuerknopfes kann die Antwort eingegeben werden, die mit der ENTER-Taste beendet wird. Falls das eingegebene Wort die richtige Länge hat, wird es in das Rätsel eingesetzt. Kann man die Frage noch nicht beantworten, gibt man eben ein Wort falscher Länge ein und geht zu einem anderen Pfeil. Soll das Rätsel für später abgespeichert werden, gibt man einfach als Antwort auf eine belie-

BÖRSE

Verk.: Modul xb (100,–; Schach (50,–); Stat (50,–) Rechnungsstellung (50,–) Lagerverw. (50,–)/Rec. + Kab. (100,–)/ ti-contr. + dskmanager I (250,–)/
* T. Schramm, 040/72 11 882
040/72 43 787

+++ TI-99/4A Kons. Mit Zubehör; Video out+WxtBas+ 32 K + Joyst + Rec Kab + 5 Bücher (ED/AS in dt. TI-Rev + div. Softw. von Spiele bis Debug auf Kass. + nur zus. + Gebot an Gerhard Biehl, Hinterstraße 8, 2209 Herzhorn oder nach 20 Uhr 04124-7072.

Verkaufe für Peribox: 32 K DM 250,– ED/ASS DM 100; RS 232 mit Kabel DM 250; Speech-Synth. DM 100; Tel. 02136/34170.

Verkaufe LOGO-Programm (XB, 32 kB) mit 30! Befehlen, Prozeduren und Variablen!! Ausf. Anl. Schickt DM 20 an M. Bannert, Mozartstraße 5, 8676 Schwarzenbach, Telefon 09284/8338.

Verkaufe viele Bücher und Joyst-Adapter (DM 25). Suche billige Module + Graphiktableau + MC-PSLOT-Routine (für XB). Zahle gut (??). M. Bannert, Mozartstraße 5, 8676 Schwarzenbach, Telefon 09284/8338.

Verk. Kabel f. ext. 32 kB – GP-250X. Bauanl. audio Ausg. nach Stereoanlage. Suche: TI-99/4A + ExBa. (plus II?) + Exp.-BOX m Contr. + 1 Laufw. + E/A-Modul m. dt. Anleitung + Maus + f. Speech-Editor und Synthesizer Handbuch (evtl. z. ausleihen) Kontakte zu TI-Usern aus Plz. 3000.. Angebote an P. Strohmeyer, Haltenhoffstr. 181, 3000 Hannover 21.

Ihr habt Probleme mit eurem TI? Egal welche. Wendet euch an die Leute mit dem größten Know-How. TI-99er Workshop Rheinland. Telefon 0221/03979!!!

An alle TI-User im Rheinland! Wenn Ihr Kontakte zu einem Club sucht, kommt zu uns. Wir sind für alle offen. Beitrag 0,00 DM TI 99er Workshop Rheinland. Telefon 0221/703979, ab 19 Uhr.

LISTINGS

```
10 ! **********************
11 ! *                    *
12 ! *  KREUZWORTRAETSEL  *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *  Burghardt Knedel  *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *      Joystick      *
23 ! * Cassetten-Recorder *
24 ! * oder Disk-Laufwerk *
25 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     5998 Bytes     *
28 ! *                    *
29 ! **********************
100 CALL CLEAR :: DISPLAY AT
(1,7):"kreuzwortraetsel" ::
DISPLAY AT(8,4):"burghardt k
nedel  2/85" :: DISPLAY AT(1
1,4):"benutze joystick 1" ::
 DISPLAY AT(13,4):"loese alp
halock taste"
110 DISPLAY AT(19,1):"fuer r
aetsel speichern bitte" :: D
ISPLAY AT(21,1):"<speichern>
 eingeben" :: DISPLAY AT(23,
1):"cassette oder diskette"
120 DISPLAY AT(24,1)BEEP:"c
oder d" :: CALL SCREEN(12)
130 CALL KEY(0,TA,K):: IF K=
0 THEN 130 ELSE IF TA<>99 AN
D TA<>100 THEN CALL SOUND(12
0,440,1):: GOTO 130
140 IF TA=99 THEN LL$="CS1"
ELSE LL$="DSK1.KREUZ"
150 OPTION BASE 1 :: DIM A$(
23,28),U$(24):: CALL MAGNIFY
(2):: FOR I=1 TO 28 :: CALL
SPRITE(#I,63,RND*16,100,125,
RND*20-10,RND*20-10):: NEXT
I
160 CALL COLOR(1,14,1,8,14,1
):: FOR I=65 TO 90 :: CALL C
HARPAT(I,X$):: CALL CHAR(I+3
2,X$):: NEXT I
170 CALL CHAR(88,"007E7E7E7E
7E7E",89,"080CFEFFFFFE0C08",
90,"3C3C3C3CFF7E3C18",125,"E
7C3810000081C3E7",33,"0101010
1010101FF")
180 DISPLAY AT(23,1)BEEP:"ge
speichertes raetsel.......ei
nlesen..<j oder n>"
190 CALL KEY(0,K,KK):: IF K=
106 THEN 740 ELSE IF K=110 T
HEN 200 ELSE 190
200 DISPLAY AT(1,1):"....XZZ
XXZXZXZXXZXZZXZXXZX.....XX!!
ZZ!X!X!ZX!Z!!Y!!!!Z.....XZ!!
!!!Y!!!!!!!!!X!ZX!!.....Y!!!
!!!Y!!!!Y!!!!!!!Z!!"
210 DISPLAY AT(5,1):"...X!Y!
!!!!Y!!!Z!!Y!!!!!!.....X!Z!
Z!!Y!!Y!!!!!!!Z!!Z!.....Y!!!
!!!Y!!!!!!Y!!!!!!!!.....XX!Z
!Y!!!!!!!!!!!Y!!!!!"
220 DISPLAY AT(9,1):"...Y!!!
!!!!XXXXXXXXXZ!!!Z!.....XZ!!
ZX!XX??????Y!!!!!!Z.....Y!!!
!!ZZX??????XY!!!Z!!.....X!Y!
!!!!X??????XX!!Z!!!"
230 DISPLAY AT(13,1):"...X!Y
!!!!!X??????Y!!!!!!!.....Y!!
!!!!!XZXXZXZXZ!Z!!Z!.....XXX
Z!Z!!Z!XX!Y!!!!!!!!!.....XY!
!!!!!!!!Z!X!Y!!!!Z!!"
240 DISPLAY AT(17,1):"...XY!
!Y!!Y!!Y!!!!!!!!Y!!!.....Y!!
!!!Y!!!!!!Y!!!!!!!Z!.....Y!!
!Y!!!!ZXY!!!!!Y!!!!!.....XXY
!!!!Z!!XX!XY!!!Y!!!Z"
250 DISPLAY AT(21,1):"...Y!!
!Y!!!Y!!!!Y!!!!!!Z!!.....XXX
Y!!!!!!!!!!!!Y!!!!!.....XY!
!!!!!!!Y!!!!!!Y!!!!!" :: CAL
L HCHAR(24,3,32,28)
260 READ A,B,P$ :: IF P$="eo
f" THEN 270 ELSE A$(A,B)=P$
:: GOTO 260
270 CALL DELSPRITE(ALL):: CA
LL MAGNIFY(1):: Z,ZS=1 :: S=
6 :: SS=41
280 CALL DELSPRITE(#1):: CAL
L JOYST(1,X,Y):: CALL KEY(1,
K,T):: CALL SPRITE(#1,125,2,
ZS,SS):: IF K=18 THEN 310
290 Z=Z-SGN(Y):: ZS=ZS-SGN(Y
)*8 :: S=S+SGN(X):: SS=SS+SG
N(X)*8
300 IF Z<1 OR Z>23 OR S<6 OR
 S>28 THEN 270 ELSE 280
310 CALL GCHAR(Z,S,XX):: IF
XX<>89 AND XX<>90 THEN CALL
SOUND(400,-6,0):: GOTO 280
320 CALL SOUND(300,440,0)::
X$=RPT$(" ",28)&A$(Z,S)&" "
:: FOR I=1 TO LEN(X$):: DISP
LAY AT(24,1):SEG$(X$,I,28)::
 CALL KEY(1,K,T):: IF K=18 T
HEN 340
330 NEXT I :: GOTO 320
340 DISPLAY AT(24,1):: ACCEP
T AT(24,1)BEEP:C$
350 IF C$="speichern" THEN C
ALL SOUND(500,440,1):: GOTO
710
360 IF XX=90 THEN 380
370 GOSUB 670 :: FOR I=1 TO
```

```
LEN(C$):: CALL HCHAR(Z,S+I,A
SC(SEG$(C$,I,1))):: NEXT I :
: GOTO 280
380 GOSUB 650 :: FOR I=1 TO
LEN(C$):: CALL HCHAR(Z+I,S,A
SC(SEG$(C$,I,1))):: NEXT I :
: GOTO 280
390 DATA 1,8,weltorganisatio
n,1,9,geradlinig,1,12,traurig
machen,1,14,verzueckung,1,
16,dunstglocke,1,19,islam.:j
esus
400 DATA 1,21,afrik.fluss,1,
22,fuer¦je,1,24,kuechengeraet
,1,27,dreist¦vorlaut
410 DATA 2,10,bibl.stammvate
r,2,11,ruhig,2,17,altes schl
echtes pferd,2,20,engl.langs
am,2,23,gr.beutel,2,28,beiga
ben
420 DATA 3,7,halt,3,13,krimi
nalbeamter,3,25,verneinen
430 DATA 4,6,gedichtform,4,1
3,altgriech.philosophenschul
e,4,18,milde gabe,4,26,moder
ner flugkoerper
440 DATA 5,8,arabisches fuer
stentum,5,15,wurfspiess,5,19
,lat.sache,5,22,offizier
450 DATA 6,8,teil von vietna
m,6,10,grossmutter,6,13,spie
lkartenwert,6,16,deutscher d
ichter,6,24,einschuechterung
460 DATA 6,27,abk.fuer ausse
r dienst
470 DATA 7,6,losung,7,13,him
melskoerper mz.,7,20,verkauf
skartell
480 DATA 8,9,maennl.vorname,
8,11,muerrisch¦grimmig,8,23,
schiffsliegeplatz
490 DATA 9,6,conferencier,9,
23,blaetterteighoernchen,9,2
7,haeufiger name fuer papage
ien
500 DATA 10,7,liebesgott,10,
10,perser,10,21,ehem.dt.bund
espraesident,10,28,dt.opernk
omponist
510 DATA 11,6,eifrig,11,12,r
oem.feldherr v.chr.,11,13,ka
eufer,11,22,panikart.ansturm
,11,26,russ.maennername
520 DATA 12,8,spitzensportle
r,12,25,maennl.bibelgestalt
530 DATA 13,8,kreisstadt in
hessen,13,21,autobahn der us
a
540 DATA 14,6,schiffsrippen,
14,15,fest,14,18,jaemmerlich
,14,20,seebad in florida,14,
22,aufsehen,14,24,heldengedi
cht homers
550 DATA 14,27,arab.vater
560 DATA 15,9,einwohner groe
nlands,15,11,sauberkeit,15,1
4,kostspielig,15,19,bekehrer
 von nichtchristen
570 DATA 16,7,treuebrecher,1
6,17,roem.zahl 1500,16,21,ju
nges rind,16,26,darsteller t
raumschiff steward
580 DATA 17,7,engl.ist,17,10
,persoenl.fuerwort,17,13,fra
nz.und,17,16,ind.fuerstin,17
,25,stadt in vietnam
590 DATA 18,6,herrschaftl.di
ener,18,12,last,18,19,aufste
llung einer maschine,18,27,g
ewaltig
600 DATA 19,6,einfarbig,19,1
0,benennung,19,15,schriftstu
eck,19,17,teil einer kette,1
9,23,prophet
610 DATA 20,8,nahrungsmittel
,20,13,finn.hafenstadt,20,20
,zeitabschnitt,20,24,zufluss
 der donau,20,28,rumaenische
 muenzen
620 DATA 21,6,papstname,21,1
0,griech.buchstabe,21,14,sch
nepfenstrauss,21,19,kleines
enges zimmer,21,26,skatbegri
ff
630 DATA 22,9,gewuerz und te
epflanze,22,22,trockenheit
640 DATA 23,7,traubenzucker,
23,16,stadt an der ems,23,23
,tropische harzart,1,1,eof
650 FOR J=1 TO LEN(C$):: CAL
L GCHAR(Z+J,S,X):: GOSUB 700
660 NEXT J :: CALL GCHAR(Z+J
,S,X):: IF X=33 THEN 690 ELS
E RETURN
670 FOR J=1 TO LEN(C$):: CAL
L GCHAR(Z,S+J,X):: GOSUB 700
680 NEXT J :: CALL GCHAR(Z,S
+J,X):: IF X=33 THEN 690 ELS
E RETURN
690 CALL SOUND(800,110,0,-4,
0):: DISPLAY AT(24,1):"das w
ort hat falsche laenge" :: F
OR I=1 TO 700 :: NEXT I :: D
ISPLAY AT(24,1):: GOTO 280
700 IF X=88 OR X=89 OR X=90
OR X=32 THEN 690 ELSE RETURN
710 FOR ZZ=1 TO 23 :: FOR SS
=6 TO 28 :: CALL GCHAR(ZZ,SS
,X):: U$(ZZ)=U$(ZZ)&CHR$(X):
: CALL HCHAR(ZZ,SS,63):: NEX
T SS :: NEXT ZZ
720 OPEN #1:LL$,INTERNAL,OUT
```

LISTINGS

```
PUT,FIXED 192
730 FOR I=1 TO 23 STEP 8 ::
PRINT #1:U$(I),U$(I+1),U$(I+
2),U$(I+3),U$(I+4),U$(I+5),U
$(I+6),U$(I+7):: NEXT I :: C
LOSE #1 :: CALL CLEAR :: END
740 CALL CLEAR :: OPEN #1:LL
$,INPUT ,INTERNAL,FIXED 192

750 FOR I=1 TO 23 STEP 8 ::
INPUT #1:U$(I),U$(I+1),U$(I+
2),U$(I+3),U$(I+4),U$(I+5),U
$(I+6),U$(I+7):: NEXT I :: C
LOSE #1
760 CALL CLEAR :: FOR I=1 TO
 23 :: DISPLAY AT(I,4)BEEP:U
$(I):: U$(I)="" :: NEXT I ::
 GOTO 260
```

BÖRSE

Verkaufe billig! Wie neu! Für TI 99/4A Bücher Tips + Tricks u. Bas/ExBas. für Anf. u. Forgeschr. (TI), Lehrcas. Basic u. ExBas., TI-Revue 1-8, TI-Revue Spez. 1+2, 10 Grafik Prgm's (Cass.) Anrufen lohnt sich! 0231/ 87 59 16 ab 18 Uhr

---

Verkaufe:
32k-Erw.-intern 250 DM, TI-Ex.-Bas. 150 DM (incl. engl. Handbuch), Centronics-Kabel f. FX80 60 DM, Kass. Rec.-Kabel 20 DM, Buch Special I 20 DM.
Tel. 02821—18 399

QUBIE & THE CUBES
Superspiel f. TI 99/4A (ex basic, ähnl. Q-Bert), 3D-Graphik, DM 25,— incl. Prgm-cassette, Porto und Versand Softwareinfo gg. DM 1,— P. Rieger, Talstr. 64, 6750 Kaiserslautern

---

Verk. TI 99/4A Neu 190 DM+CsKb 18 DM + 2 TI-Contr. 40 DM + Org. Module: Text/Dateiverw. + Analyse + Statistik a 55 DM + Invader + Parsec + Zero Zap + Othello + a 35 DM + Schach 55 DM + Lerncas., Basic + Ext. Basic a 10 DM + Tips Tricks + Spiel/Lern/Arbeite m. TI a 25 DM.
Tel. 07773-5261

Pläne für TI-99/4A anschl. US Kabel (Disk Reset, Akustikkoppler Drucker I-Sticks) E. Mauk,Gollierpl. 10, 8000 München 2

Suche dringend Schach-Modul für TI99/4A
Zuschriften an Mittermair, Johann Osternbergerstr. 57 A-5280 Braunau/Inn

---

Suche
RS 232-Karte für Box dt
E/A Handbuch dt TI-Writer
Anleitung Programmroutinen 2&3 (dt)
Uli Warias, Tel. 05300/485

Biete prgr. jgl. Art: Mathe/ Statistik/El. Technik/Graf. Spiele/Finanzen (Hasuhalt) Erledigung von Aufträgen Rechner: MSX, Atari, ZX81 Thomas Wächter, Mühläckerstr. 30, 7312 Kirchheim-Teck

---

Kaufe Mini Memory Modul 80—100 DM je nach Zustand / Spiel-Module 15—25 DM je nach Zustand
Tel. 02652/61 57

Verk. TI Konsole + 2 Joysticks + Recorder + TI Tips + Tricks + Listings VB 370 DM, B. Maack, 3074 Steyerberg

---

99/4A Konsole + Exbasic je DM 100,—, Adu-Modul: DM 40,—, 25 Progr.-Cass. Atari-Module für TI je DM 50,— (Pole-Position, Moon-Patrol, Shamus, Popeye, Miner2049)
TI-Module: (Treasure Island, Microsurgeon je DM 45,— Fathom DM 35,—)
R. Schwalbe 02368/6237

TI-Freunde hergehört!
Der Clut der Toten bietet:
2099er
Eine richtige Computerzeitung für TI-99/4A und VC-20. Mit Listings, Tips, Berichten, usw auf mehr als 40 Seiten. Anforderbar gegen zwei 50 Pfg. Briefmarken für Porto und Verpackung bei:
2099er, Am Hang 5,
2054 Geesthacht

Biete/Suche + tausche Hardw. einzeln/kompl. sowie org. Softw. ab 17.00 05725/ 6409 G. Buerger, 3051 Auhagen

---

TI99/4A + Cass. Rec. + 20 z.T. Besp. Cass. + TI-Revue kpl. + ca. 100 Listings + 2 Joyst. + Kabel komplett 450,—. Tel. 069/49 55 01

VE. 8K7 Dynamics. Lager Kartei, Data Bank, Adres. Büch. Archiv usw. + 3 Kas. TI Adventure. TE DM 18. incl. Porto. J. Dhulst, 40, R. des Deportes, B 6478 - Rance, Belgie.

Umfangr. TI 99/4A System, Box m. 32K, Schnittst., Floppy, 2. Floppy ext., Sprachsy., Text-Datei-3D-Graf. auf Disk, mehr. Module u.v.m. evtl. auch einz. zu verk Näheres unter Tel. 07365/ 6235 n. 18 Uhr

---

Verk. TI99/4A + Ex. Basic, engll. Handb. kpl. auch einzeln. Jungle Hunt, Amazeing, Munchman, The Attack, Tombstone City, Carwars + ca. 200 Prgm. + Joyst. + R-Kabel + Basic Lerncass. + Bas./Ex. Lehrbuch. VP: DM 600,— Tim Dinter Tel. 08191/2885

*** Verkaufe kompl. Station 99er + PBox + 1 Disintt + 1 x Diskext + 32 k + I/O-Karte + Monitor (grün, 20 MHz) + Drucker (Epson TX-801 + Palm od. Joystick (TI + Spect) + EdAss + TI-Writer + Ex-Basic + PRK + 6xModule + Forth + ca. 700 Pgm auf 100 Disks (ca. 20 % Masch Code) + Service-Unterlagen + alle TI-Journal + -Revue. VB 3450,— DM
J. Zander, Kufbergerstr. 20 8411 Wenzenbach
09407/2010

TI99/4A
Disklauf * Controler Divers. Software * Divers. Module * Preis VB * einzeln *
Tel. 0871/25791

Verkaufe TI99-4A, Exbasic, EA-Modul, Peribox mit Laufwerk, 32k Erw., RS232, TI-Writer (dt) Multiplan, Recorderkabel 2-fach, TI-Joyst., Speechsynth., div. Disks, Preis kompl. DM 1000,— VB. J. Schneiderat Mo-Fr 9-15, 0211-632042

Wir kaufen Ihre defekte TI-Konsole und Peripherie zu Höchstpreisen, rufen Sie an, bevor Ihr TI in den Müllsack wandert. Bei Brose 02303/65134

TI99/4A + Ext. Basic + Rec. Kabel + Joyst. Adapter + Flugsimulation + Tips & Tricks von Data Becker + Programme für den TI von R. Heigenmoser, kaum benutzt, für 380,— DM
Tel. 040/79 04 110

Verk. origin. TI Dsk-Contr. + Dsk-Manager: 300 DM. C. Kater 7, R. de Schoenfels L-7432 Gosseldange, Tel. 328060

TI-99/4A, Ex-Basic, 2 Joyst. 2 Spilmod., Kassetten, TV Netzkabel, Bedienanleitung 8 TI Revue gegen Gebot zurück
Helmut Weiler, Freiherr von Stein 1 , 6148 Weiterstadt

TI-Bücher zu verkaufen a 15 DM. Tips & Tricks, Spiel Lern. Arb. Ex-Basic engl., Farbe. Grafik. Ton Spiele, Kcohbuch I und II Spielprogramme selbst erstellen I un und II. (insg. f. 100 DM). Peter Hielscher, Am Wall 22, 4401 Saerbeck

---

Original II-Kass. + Module zu verkaufen:
Grundkurs 10, Finanzbeater 20, Market-Planspiel 20, Vier gewinnt 30, Video II 30, Adv. 50 Hangman 30, Amazeing 30, Hustle 30 (insg. 200). Hielscher, Am Wall 22, 4401 Saerbeck

---

Suche Sprachsynthesizer
Verkaufe Exteeded-Basic Modul mit deutschem Handb.
(04554) 63 96

Verkaufe wg. Systemw. TI-99/4A + PBox + RS232 + 32K + Disk + Contr + ExBas + Mod + Lit VB 1700 DM ab 18 Uhr. Tel. 04921/ 27319

Suche Erweiterung m. Centronics und Druckerkabel sowie andere Geräte z. Erweiterung und Anwenderprogramme.
H. Dettling, Ottostr. 8, 5061 Beg. Bladb. 1

Verk. orig. TI-Laufwerk 200,— DM, orig. TI-Joystick 40,— DM, Minimemory mit Büchern 180,— DM
08131/93046

HILFE! Wer verkauft mir defekten TI? Peter Kliem, Elsa Brandströmstr. 37, 5042 Ergtstadt

---

Ext. 32K CMOS: 200 DM
Ext. Centronics: 100 DM
Tausche auch gegen die entspr. Karten! Tel. 04171-71841 / Axel

---

BÖRSE

Verkaufe TI 99/4A + Ext-B. + Cass.-Kabel + orig.-TI Joysticks + Prog.-Kassetten + Literatur DM 400. Tel. 0541/18 75 71 ab 18.00

Verkaufe: TI99/4A + Kas.Rek. + Kabel + 5 Module + 2 Joystick + Handbuch + 7 TI Revues VB 400 Tel.: 06107/4128

---

* HELP *
Suche Pole-Position, Moso Pacman, Donkey Kong, Buck Rogers, Jungle Hunt, Espial oder Star Treck. Angebot bitte an: Manfred Lipowski In der Wanne 165, Castrop-Rauxel 4, Tel. 02305/72237

Programme: alle in Ex-Basic Liste gegen Rückporto anfordern. V. Brose, Hamburgerstr. 1, UNNA

Suche TI-Writer, E/A-TI-Logo-Modul, P-Code-Karte + Handbücher (deutsch) H. Fricker, Trachselweg 15, CH-3008 Bern, Tel. 031/ 45 25 91

Verk. TI 99/ 4 A + XBasic + Rec. Kabel + Basic Lernkas. + Data-Becker BVCH für nur VB 250,--. Tel. (05242) 36146.

Verk. Org. Adventure-Kas. Neu – Orf. verpackt – Einz. 10,--, Rabatt bis 10 St. 60,-- W.-Reinders, Postf. 6613, 4000 Ddf., Tel.: 0211/ 683615.

Um meinen TI 99/4A zu erweitern, verkaufe ich mein Phillips Telespiel mit 8 Module VB 295,--, T:05062/8089.

Suche diverse Hard- u. Software-Angebote (Listen etc.) an Michael Wank, Hauptstr. 60, 5204 Lohmar 1.

Suche Pal-Modulator für TI, evtl. mit defekter Konsole. Gerald Sude, Tel.: 0711/ 379431 ab ca. 19 Uhr.

– – – VERKAUFE – – –
Buch in deutsch DM 40. TI-Basic und X-Basic für Anf. und Fortgeschr. Tel.: 089/ 1572559.

– – – VERKAUFE – – –
XB-Utilities DM 70,--. Assembler Kurs Asem 4 von Ida, DM 70,--. Tel.:

TI 99/4A PBox + E BAsic + ED Assemgl + Diskkontr + 32K + RS 232 + Laufwerk + Modul + PROYR etc. Tel.: 06431/22857 ab 19 Uhr.

C-64 ACHTUNG C-64
Verkaufe über 100 Spielepokes zum Preis von nur 5 DM in bar. Schreibt an: F. Marth, 68 Rue de L'Eglise, 7224 Walferdingen, Luxemburg. Garantierte Antwort.

Suche für C-64/1541 Compiler mit Unterl. zB. Data Bekker: Basic 64/o.ä. – Esoterik-Horoskop/Bio/Kabbala/ Wahrs./Tarot usw./Ferner: Service-Unterl. f. SX-64/ Hard-tools f. SX-64& + Soft-Tools. Fred Seyferth, Balger Hauptstr. 10, 7570 Baden-Baden. 07221/65240.

Suche: TI-Kontakt. Raum buchen, Ext.-Basic (II plus?), P-Box m. 32 K, Disc.-Kontr. Laufwerk, Module, Bücher. 062921326.

TI 99: Verkaufe Cassettenrecorderkabel: 19 DM, Fachbuch "TI 99/4A Intern" (neuwertig): 29 DM und gebrauchte TI-Ersatzteile. Holger Wenzel 06138/8395.

Verk. TI 99/A4 + X-Basic + Org. Diskstation + Disccontr. + CA 80 Spiele, Preis: 700,-- DM. An Stefan Kuttner, Tel: 08704/670.

Verkaufe TI 99/4A mit 500 Programme + 13 TI Hefte + Joystickadapter 300,-- DM. Telefon: 08971/9835.

Suche f. TI 99/4A Ext. Disc-Contr. + Floppy bis 450 DM. Verkaufe Sprachsynth. f. 100 DM. A. Schlaht/ 04205/ 1857.

Verkaufe für TI 99/4A Original TI-Soysticks und Assemblerkurs ASEM-4 in dt. Sprache auf Diskette incl. 2 Bücher, lauffähig in Ex. Basic. Preis = VHS. Tel.: 0234/ 382596.

Verkaufe für TI MBI-Centronic-Schnittstelle, VB 100,--. Torsten Meyer-Staufenbiel, Brucknerweg 4, 7257 Ditzingen 5.

TI 99/4A Computer mit Spiel Modul, 190 DM. Recorder, Kabel 20 DM. Farbe Fernsehen, 52 cm, 350 DM. Ed/Ass-Buch 50 DM, alles zusammen 510 DM. Tel: 06155-4692.

Billig abzugeben: Neuer Matrixdurcker Seikosha GP 100 A/Mark 2 mit Kabel für Centronicx-Schnittstelle, Werksgarantie, Preis Ö.S. 2.300,-- Wien 10, Tel.:

Wirkl. neuw. 99/4A (ca. 5 Betriebsst.) + Rec. + Kabel, inkl. Handb. u. Tips & Tricks für nur 199,-- zu verkaufen. Ab 20 Uhr 08031/81345.

TI + Rec + XBasic = massig viel Zubehör für 600 DM oder Sinclair Spectrum mit Drums ynthi = Wertausgleich. Armin Holz, Rathausstr. 38, 5014 Horrem.

Kaufe Spiel-PRG F. TI in TI-B und Ext-B-Liste an: D. Boeschen-Kraue, FR-Ebertstr. 11, 2890 Nordenham.

Verkaufe TI-Ex. Basic + Flugsimulatro + Digger für nur 150 DM. S. Schmidt, Laatzen 22, 3258 Aerzen 2.

Terminal Emulator 1200. Suche Beschreibung! W. Stresemann, Gatowerstr. 83c, 1000 Berlin 20.

TI-99- 4A + X-Basic + Datenverw. + Schach + Spiele + CC mit Kabel + Literatur + Software VB 450,--. 08461-1404.

E/A Handbuch sucht Tel.: 0211/429627.

Suche: Joystic ca. 20,--. Modul: Compact Engl. + Rechenkuenstl. 07483/221, Horst Fischer.

Suche preisw. Original TI-RS 232, Modul-Expander u. Ex-Basic II . 18 – 20 Uhr, Tel.: 04331/6471.

---

32 K – Extern - Direkt – Ansteckbar – Akkuger. 225,-- DM, Original TI-Laufwerk 250,-- DM. Tel.: 06181-81848.

Orig. TI-RS232 Schnittstelle extern, Bus durchgeführt und Kabel nur 210,-- DM. Adventuremodul + 11 Cass. nur 110,-- DM. Tel.: 05021/ 13700.

Suche Disk-Controller oder intern) für TI ii)4 A. Angebote an: M. Wechsung, Goethestr. 8a, 3300 BS.

TI-User sucht DFÜ Kontakte, Karte genügt und an Harald Vöge, Pergamentweg 44b, 2000 Hamburg 74.

Der TI-Club Hamburg sucht weitere aktive. Tel.: 7128586 + 2994609.

Suche Diskcontroller intern für TI 99/4A-Peri-Box Tel.: 030/6638779

32-K Erweiterung, extern am 1/0 Port mit durchgeführtem Bus für 150,-- DM incl. Gehäuse. 030/6638779.

Die Spiele in Basic: Ghostbusters 10,-- DM, Raid over Moscow 5,--, Frogger 4,-- + PRG-Liste + MC + Porto nur g. bar. Stephan Kamp, Auberg 4, 5450 Neuwied 13.

Suche gute deutschsprachige TI-Adventure! Biete je Prgr. 3 Programme nach Wahl aus meiner Progr.-Sammlung !). Verkaufe schwarzes Orig. Ext.-Basic-Modul incl. ca. 50 Ext. Progr. + Data-Becker Buch + ExBas-Lehrg. (8 Progr. D). T: 02174/40654.

Suche Adventure Modul für 40 DM. Tel: 02236/41342.

Org. Module mit Handbuch: Donkey Kong 40,–, Defender 30,--, TI-Invaders 20,--, Zero ZAP 20,--, Othello 20,--, Align Addition/Minus Mission je 10,--, Basic/Ex-Basic Tutor je 10,-- . Tel: 0561/18967 od. 496351.

Verk. TI 99/ 4A + Rec. Kabel + X-Basic m. dt. + engl. Anleitung, Preis: 400 DM. ZX 81 TV-Modul defekt und 16 K-Erweiterung m. engl. Handbuch, VB: 100 DM. E Schuermann, Sonnenstr. 40, 5940 Lennestadt 11, Tel.: 02721/ 10871.
Originalverpacktes Extender-Basic Modul mit deutschem Handbuch, Preis: 165 DM. (04554) 6396.

Tausche Masch. Prg. und auch andere Softw. Max Nagler, Tradtstr. 14, 8492 Furth i. Wald, Tel.: 09973/1517.

Suche: Schachmodul o. Parsec (Tausch gegen Attack) M. Nagler, Tradtstr. 14, 8492 Furth, 09973/1517.

Suche für TI 99/ 4 A Disketten-Laufwerk und Drucker mit entsprechenden Anschlüssen. Tel.: 0461/51509.

Verk. TI 99/ 4 A, neu: 190 DM + Kabel 15 + 2 Joyst. 40 + Module: Text/Datei, verw. + Analyse + Statistik, je DM 50,-- + Invaders. Walter Freidinger, Kronbühl 20, 7762 Bodm.-Ludwigshafen.

TI 99/ 4A + ex. Bas. + Handbuch + Rek. Kabel + 11 TI-Revue + Lit. + Lehrcass. + DataBank + Lagerkartei = DM 600,--. Tel.: 0451/ 593394.

---

albs IMMER NEUESTE HARDWARE FÜR DEN TI-99/4A ..... DIREKT VOM HERSTELLER

EXTENDED-BASIC (Mechatronic) mit deutschem Handbuch 198,–

EXTENDED-BASIC II PLUS mit deutschem Handbuch = Extended-Basic + Grafik Extended-Basic (Apesoft) in 1 Modul 285,–

Umtauschaktion
Bei Bestellung eines EXTENDED-BASIC II PLUS vergüten wir Ihnen DM 70,– bei kostenfreier Zusendung eines original amerikanischen Extended-Basic-Moduls (elektrisch/mechanisch einwandfreier Zustand!!!) Sie zahlen nur noch 215,–

Umbauaktion (gilt nur für deutschen Lizenznachbau „Mechatronic"). Wir machen aus Ihrem EXTENDED-BASIC ein EXTENDED-BASIC II PLUS mit deutschem Handbuch für nur 98,–

32-k-RAM-ERWEITERUNG mit Centronic-Interface, Kunststoffgehäuse 190 x 110 x 60 mm zum seitlichen Anstecken an den Bus, der Bus wird nach rechts durchgeschleift, mit 5-V-Steckernetzteil 289,50

Unser Paketpreis-Angebot
EXTENDED-BASIC II PLUS + 32-K-RAM-ERWEITERUNG ohne Centronic-Interface für nur 499,50

128-k-RAM-ERWEITERUNG, mit Centronic-Interface und 5-V-Steckernetzteil 399,50

Technische Änderungen vorbehalten
Versand gegen Nachnahme oder Vorauskasse

4-FARBEN-PRINTER-PLOTTER-A 4, Centronic-Schnittstelle, DIN-A 4-Format, Direktanschluß an 32-k- oder 128-k-RAM-Erweiterung 699,–

ANSCHLUSSKABEL von 32-k- oder 128-k-RAM an PP-A 4 68,–

SLIM-LINE-LAUFWERK 5.25", 500-k-Byte-DS/DD (z. B. TEAC FT 55 B) 399,90

EINBAUSATZ für 2 Laufwerke in original TI-P-Box 95,–

DISC-STEUERKARTE (CorComp), DS/DD, für max. 4 Laufwerke 635,–

Die Weltneuheit: GRAM-Karte für die P-Box mit 128k eröffnet völlig neue Pespektiven; u.a. Entwicklungssystem für eigene „Module". 698,–

TI-MAUS – die schnelle und komfortable Cursorsteuerung mit Software auf 5,25"-Diskette, mit 5-V-Steckernetzteil 296,–

EPROMMER – zum Brennen von EPromms bis 27128, alle Funktionen Software gesteuert, Programm auf 5 1/4 Zoll Diskette mit Steckernetzteil 278,90

TI 99/4A Intern von Heiner Martin (englische Ausgabe) 34,95

albs-Alltronic · B. Schmidt · Postfach 1130
7136 Ötisheim · T. 07041/27 47 · Tx. 7 263 738 albs

Zeig beim Porto Herz & Verstand:

Kauf Wohlfahrts-briefmarken.

Hilfe, die ihr Ziel erreicht.

Erhältlich bis Ende März bei der Post, ganzjährig bei den Wohlfahrtsverbänden.

IMPRESSUM

TI-SPECIAL
erscheint in der AKTUELL-GRUPPE, Elisabethstraße 1, 8044 Lohhof.
Redaktion: Senator-Presse-Service.

Verantwortlich für den Inhalt: Heiner Martin.
Verantwortlich für Anzeigen: Bruno Redase.
Verantwortlich für Listings: Harmut König.
Alle: Postfach 1107, 8044 Lohhof. Anfragen bitte nur schriftlich.

Druck: Maier und Söhne
Es gilt die Honorarliste des Verlages. Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung.
Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für einen einmaligen Abdruck sowie die Aufnahme in den Programm-Service nach den Verlags-Sätzen! Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Jedwede Verwertung ist untersagt. Nachdruck nur mit ausdrücklicher schriftlicher Zustimmung des Verlages.

Namentlich gezeichnete Artikel geben nicht unbedingt die Meinung der Redaktion wieder.
Kein Anspruch auf Lieferung bei Ausfall durch höhere Gewalt.
Gerichtsstand: München
Geschäftsführer: Werner E. Seibt
Abo- und Kassetten-Service: Henny Rose Seibt

© by TI/CBM Verlag SPS und Autoren.

Die nächste TI-REVUE 25.April 86

QUALITÄT ZU KLEINEN PREISEN? DANN KOMM ZU KOLLASKE!

Claus-Dieter Kollaske
24-Stunden-Service
Computer Hard- and Software
Grazer Damm 223
1000 Berlin 41
Tel. 030 / 796 83 71

*** APRIL ANGEBOT **********
********* TI 99 / 4A **********
**********UNSER PREIS *****
********** 329,– DM **

Wir lassen den TI-USER nicht im Stich!

atronic

Hardware-Software-Zubehör
Das komplette Angebot für den TI 99/4A-Besitzer

Hardware
CPS 99, Disk-Station
Speichererweiterungen
Schnittstellen
Disk-Controller
Drucker

Software
Professionelle und Spiele-Software

Zubehör
z.B. Disketten
Drucker-Papier
Kabel

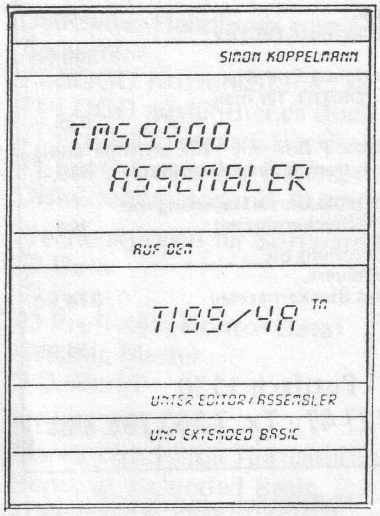
Bücher
viele Titel für den TI-Besitzer

FORDERN SIE DIE PREISLISTE AN!
— Der ATRONIC-Service löst (fast) alle Probleme —

atronic-Produkte bekommen Sie bei jedem guten TI-Händler oder direkt bei:
atronic · Meiendorfer Weg 7 · 2000 Hamburg 73 · Tel. 0 40 / 6 78 93 08-09 · Tx. 2 174 031

...ENDLICH !!!!

DAS BUCH FÜR DEN ASSEMBLER-PROGRAMMIERER



280 Seiten
geballte Information

Ein umfassender
Überblick
über TMS 9900
und TI 99/4A

Ein unentbehrliches
Nachschlagewerk

Buchpreis DM 49,-

Programm-
diskette DM 20,-

Beschrieben sind u. a. Prozessor, Assemblercode Befehle, Direktiven, CRU, DSR, GROM, VDP, TMS 9901, Tastatur u. v. m.

Buch und Diskette sind zu beziehen im Fachhandel oder direkt vom Autor: Simon Koppelmann, Judenpfad 7, 5000 Köln 50

Versand nur gegen Vorkasse (Scheck oder Überweisung auf Konto 2255488900, BfG Köln 370 101 11) oder Nachnahme zuzüglich Versandkosten: DM 4.- (Scheck/Überweisung) / 8.- (Nachnahme)

128 K GRAM-KARTE

für die TI 99/4A-Peripheriebox. Erweiterbar auf 512K!

Mit der 128 K GRAM-Karte, an deren Entwicklung wir beteiligt waren, entstand für den TI 99/4A-User das ideale Handwerkszeug für „GPL".

Quick-Information

CALL EDITHEM: Aufruf eines Hexmonitors aus dem Basic oder Extended Basic im Direktmodus, mit dem sich Speicherinhalte verändern lassen.

CALL GRAM: Aufruf aus dem Basic o. Extended Basic im Direktmodus des LOADER-Utilityprogramms, welches das Laden und Abspeichern auf Diskette von Modulinhalten gestattet.

CALL MODUL („NAME"): Umwandlung eines TI-Basic-Programms in ein „Modul", d.h. das Programm wird in das GRAM verlagert und kann dort über das Titelbild ausgeführt werden. Bringt bis zu 13 KByte mehr Speicherplatz für Variable.

LOADER-Utility: Dieses Hilfsprogramm kann einmal direkt über das Menue der GRAM-Karte oder über CALL GRAM bzw. CALL GRAM1, CALL GRAM2 und CALL GRAM3 für weitere GRAM-Karten) ausgewählt werden.

LOAD-Datei: Ermöglicht über eine Datei im Format DISPLAY, VARIABLE 80 das Laden von mehreren gespeicherten ROM/GROM-Inhalten. Einige vorbestimmte Namen von LOAD-Dateien erscheinen im Menue der GRAM-Karte und können über einen Tastendruck angewählt werden.

lieferbar Ende März. Preis ..... 698,00 DM

GPL-ASSEMBLER

für das Editor-Assembler-Modul

Gebrauch von FMT, LABELS und Berechnungen im Adreßfeld zulässig!

Dokumentation in Deutsch oder Englisch. Programm auf Diskette 99,00 DM

Ausführliche Information über die hier beschriebenen Produkte gegen 80 Pf-DIN A5-Rückumschlag. Weiteres TI 99/4A-Zubehör in unserer Angebotsliste – kostenlos anfordern!
Versand per Nachnahme (+ 5,00 DM) oder per Vorausscheck (+ 3,50 DM)

ELEKTRONIK-SERVICE
Peter Kleinschmidt Linning 37 4044 Kaarst 2 Tel. (02101) 603208

atronic DER PARTNER FÜR TI99/4A

MITMACHEN – MITGEWINNEN

Beantworten Sie folgende Fragen (Brief, Postkarte) und nehmen Sie automatisch an der Verlosung wertvoller Preise teil:

1. **Welche Peripherie (Zubehör, Software) besitzen Sie zum TI 99?**
2. **Welche Peripherie (Zubehör) planen Sie demnächst zu kaufen?**
3. **Welche Peripherie (Zubehör, Software) interessiert Sie, welche es für den TI 99/4A noch nicht gibt?**
4. **Welche Software nutzen Sie überwiegend?**
5. **Was stört Sie am TI 99/4A?**
6. **Welcher Computer interessiert Sie noch?**

1. Preis
CPS 99 mit einem Laufwerk
2. Preis
Disketten-Station mit einem Laufwerk
3. Preis
MEGA-RAM (256 KByte)
4.-10. Preis
je ein Warengutschein im Wert von 100 DM

Einsendeschluß: 15. Mai 1986
Der Rechtsweg ist ausgeschlossen

ATRONIC Produkte
Ständig neue Entwicklungen
Umfangreiches Programm
Hardware/Software/
Zubehör/Bücher
Qualität „made in Germany"

ATRONIC Service
Qualifizierte Fachberatung
Reparatur von TI 99/A4
Hardware

Falls Sie die Fragen nicht beantworten möchten, unsere neue Preisliste sollten Sie dennoch anfordern! Die Preissenkungen werden Sie überzeugen.

atronic

Meiendorfer Weg 7 · 2000 Hamburg 73
Tel.: 040/6789308 · Tlx. 2174031

TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A +

special+special+special+

TI 99/4A SPECIAL

SONDERHEFT NR. 5/86 DM 14,80

RUND 130 SEITEN LISTINGS! LISTINGS!

ÖS 124/SFR 14,80

ANWENDERPROGRAMME:

Z.B. WOHNGELD-ANWALTSKOSTEN
HAUSHALTS-FÜHRUNG-NETZTEIL-
BERECHNUNG-RESONANZ-
FREQUENZ-EAN-CODER
BÖRSEN-MASTER-MENDELS-GESETZ

SPIELPROGRAMME:

Z.B. BRD-QUIZ-DSCHUNGELJAGD-
KREUZWORTRÄTSEL-CYCLONS
STUFEN ZUM ERFOLG-RETRUN
VIER GEWINNT-HALLOWEEN

UTILITIES:

Z.B. SPEECHPROZESSOR
U.V.M.

TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A + TI99/4A +

l+special+special+speci

Zeig beim Porto Herz & Verstand:

Kauf Wohlfahrts-briefmarken.

Hilfe, die ihr Ziel erreicht.

Erhältlich bis Ende März bei der Post, ganzjährig bei den Wohlfahrtsverbänden.

IMPRESSUM

TI-SPECIAL, die Zeitschrift für den TI PC und TI 99/4A, erscheint in der
München Aktuell Verlags GmbH
Redaktion: Senator-Presse-Service.
Verantwortlich für den Inhalt: Heiner Martin.
Verantwortlich für Listings: Hartmut König.
Geschäftsführer: Werner E. Seibt
Alle: Postfach 1107, 8044 Lohhof. Anfragen bitte nur schriftlich.
Druck: Maier und Söhne
Es gilt die Honorarliste des Verlages. Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung.
Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für einen einmaligen Abdruck sowie die Aufnahme in den Programm-Service nach den Verlags-Sätzen und überträgt dem Verlag das Copyright!

Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Jedwede Verwertung ist untersagt. Nachdruck nur mit ausdrücklicher schriftlicher Zustimmung des Verlages. Namentlich gezeichnete Artikel geben nicht unbedingt die Meinung der Redaktion wieder.
Kein Anspruch auf Lieferung bei Ausfall durch höhere Gewalt.

Abo- und Kassetten-Service: Henny Rose Seibt

©by TI/CBM Verlag SPS und Autoren.

★
Gerichtsstand: München

TI99/4A

PREISSENKUNG BEI VIELEN ARTIKELN !!

Peripherie

RS 232 Karte (Original TI)	379,—
RS 232 Karte (Atronic)	299,—
32 K-Karte (Atronic)	299,—
Disccontroller DSDD (Atronic)	449,—
Compact Peripherie System CPS 99 mit 1 Diskettenlaufwerk DSDD + 10 Disketten	1399,—
CPS 99 mit 2 Diskettenlaufwerken DSDD + 10 Disketten	1749,—
Diskettenlaufwerk DSDD für Peri.-box mit Einbausatz	399,—
Externe 256 K-Erweiterung	589,—
Externe 32 K-Erweiterung	199,—
Externe 32 K-Erweiterung batteriegepuffert	239,—
Externe 32 K-Erweiterung mit 1 Centronicsschnittstelle	269,—
Externe 32 K-Erweiterung mit Centronicsschnittstelle + Druckerkabel + Epsondrucker LX 80	1169,—
dto. + Epsondrucker FX 85	1519,—
dto. + Stardrucker Gemini-10X	859,—
dto. + Stardrucker NL-10	1259,—
Sprachsynthesizer	175,—
Modulexpander 3fach	125,—
RGB-Modulator	179,—
Akustikkoppler Dataphon S 21 d + externe V-24-Schnittstelle + Verbindungskabel	539,—
Akustikkoppler Dataphon S 21 d + Kabel für RS 232 Karte	299,—
TI-Maus mit Software (Diskette)	295,—
Fernbedienung	65,—
MBX-Sprachsteuereinheit + Baseballmodul anschlußfertig	325,—
Grafiktablett Supersketch + Dig Dug + Defender + Statistik	179,—

Modulsoftware

Extended Basic II Plus	279,—
Mini Memory (Original TI)	169,—
Editor/Assembler (32 K notw.)	159,—
TI-Writer (32 K + Disk notw.)	259,—
Multiplan (32 K + Disk notw.)	259,—
TI-Logo II (32 K + Disk notw.)	299,—
Diskfixer (Navarone)	129,—
Terminal Emulator II	85,—
Connect four, Attack	NUR je 19,—
Alpiner, Chisholm Trail, Othello, Tombstone City	je 29,—
Car Wars, Defender, Dig Dug, Invaders, Munch Man, Statistik	je 39,—
Fathom, Jungle Hunt, Moon Patrol, Hopper, Parsec, Indoor, Soccer	je 49,—
Congo Bongo, Burgertime, Espial, Moonsweeper, Microsurgeon, Treasure Island, Donkey Kong, Tunnels of doom, Protector II	je 59,—
Adventuremodul, Buck Rogers, Return to Pirats Isle, Star Trek, Video Chess, Shamus	je 69,—
Popeye, Pole Position, Miner	je 79,—
Datenverwaltung + Analyse	79,—

BÜCHER

Editor/Assembler Handbuch dt.	98,—
Mini Memory Spezial dt.	98,—
TMS 9900 Assemblerhandbuch für das Mini Memory dt.	78,—
TI-99/4 A intern dt.	38,—

Disketten- und Cassettensoftware

Gesamtübersicht mit Preisliste erhalten Sie gegen Zusendung eines Freiumschlags (Kennwort: TI-99/4 A).

z.B. TI-Artist	99,—
3 D-World, Graphicmaster	je 75,—

!!! Ebenfalls im Lieferprogramm: Atari, Epson, Commodore + Schneider !!! Versandkostenpauschale (Warenwert bis DM 1000,—/darüber): Vorauskasse (DM 8,—/20,—), Nachnahme (DM 11,20/23,20), Ausland (DM 18,—/30,—). Lieferung nur gegen Vorauskasse oder per NN; Ausland nur Vorauskasse.

CSV RIEGERT

Schloßhofstr. 5, 7324 Rechberghausen, Tel. (07161) 52889

Neue Software für den TI 99/4A:

ALPHA DATECK V1.0 von U. Brüsseler, Datenverarbeitungsprogramm von privater Adressendatei bis professioneller Datenverwaltung. Gravierende Merkmale: 40. Zeichen pro Zeile — superschnelles Sortieren — Item's (Datenfelder) zu jeder Zeit änderbar — 2 Suchroutinen — superschnelle Bedienung — inkl. Datei Etikettieren. Benötigte Konfiguration: Extended Basic, 32 K RAM, mind. 1 Diskettenlaufwerk.

MONITOR V1.1 von J. Sundermann, Programm zum Generieren, Korrigieren und Testen von Assembler-Programmen. Völlig überarbeitete Fassung aufgrund Testbericht in TI REVUE 5/86, mit z.B. Setzen bzw. Löschen von Speicherbereichen — Ablegen eigener und fremder Programme ab A000 — Auslesen und Änderung der CRU-Bits — Protokollierungsmöglichkeit der gesamten Monitoraktivitäten auf Drucker oder Diskette. Benötigte Konfiguration: E/A-Modul, 32 K RAM, mind. 1 Diskettenlaufwerk.

Nicht zu vergessen: GPL-Assembler, GRAM-Karte, Public-Domain-Software für GRAM-Karte, Extended Basic II plus, ATRONIC-Produkte, Eprommer-Gerät, TI-MAUS, Adventure-Editor, TMS 9900 ASSEMBLER von S. Koppelmann, Modul-, Eprommer-, 32 K-Platinen, Spielmodule zum Ausschlachten (Schleuderpreise)

Informationen und Preise bei: ELEKTRONIK-SERVICE **Linning 37 4044 Kaarst 2 Tel. (02101) 60 32 08**

INHALT TI/SPECIAL 5/86

LISTINGS

Das Labyrinth von OX
Ein gefährlicher Weg in die Freiheit — ab Seite 4

Dschungel-Abenteuer
Finden Sie in der grünen Hölle des Amazonas die Beute einer Gangsterbande — ab Seite 10

Retrun
In einem unterirdischen Höhlensystem warten verschollene Forscher auf Ihre Hilfe — ab Seite 15

Netzteilberechnung
Der TI99 im Dienste der Lötkolbenfreaks — ab Seite 21

Resonanzfrequenz
...und noch ein Hilfsprogramm für Bastler — ab Seite 21

Vier gewinnt
Ein Strategiespiel für 1 — 2 Personen mit guter Bildschirmdarstellung — ab Seite 25

BRD-Quiz
Wie gut kennen Sie sich in Deutschland aus? Eine lehrreiche Unterhaltung für die ganze Familie — ab Seite 29

X-Format 28
Endlich für alle TI-User: Listingausdrucke im Bildschirmformat, sogar zweispaltig — ab Seite 35

Stufen zum Erfolg
Beweisen Sie in diesem Staatsführungsspiel Ihre Führungsqualitäten — ab Seite 37

Halloween
Ein Monster mit 100 Leben treibt sein Unwesen — ab Seite 45

Hasenjagd
Eine Computersimulation über das Gleichgewicht in der Natur — ab Seite 47

Die Mendel'schen Regeln
Generationen — vom TI berechnet — ab Seite 50

Speed-Racer
Der TI wird zur Rennpiste — ab Seite 53

Way to Home
Ein Grafik- und Textadventure. Finden Sie das 'Home' und achten Sie auf den Geheimgang — ab Seite 54

Commander
Der Sternenkrieg geht weiter. Zwei Spieler im Kampf um die Vorherrschaft — ab Seite 59

Wohngeld
Haben Sie Anspruch auf Mietzuschuß? Der TI99 sagt es Ihnen — ab Seite 68

Kreuzworträtsel
Der TI hilft beim Lösen und Sie können Ihre eigenen Rätsel bauen — ab Seite 75

Das Fünfzehnerspiel
Hier ist wohl einiges durcheinandergeraten. Stellen Sie die Ordnung wieder her — ab Seite 81

Flashdance
Wie gut sind Sie beim Tanzen? — ab Seite 82

Cyclons
Zwei Spieler im Feuertest — ab Seite 87

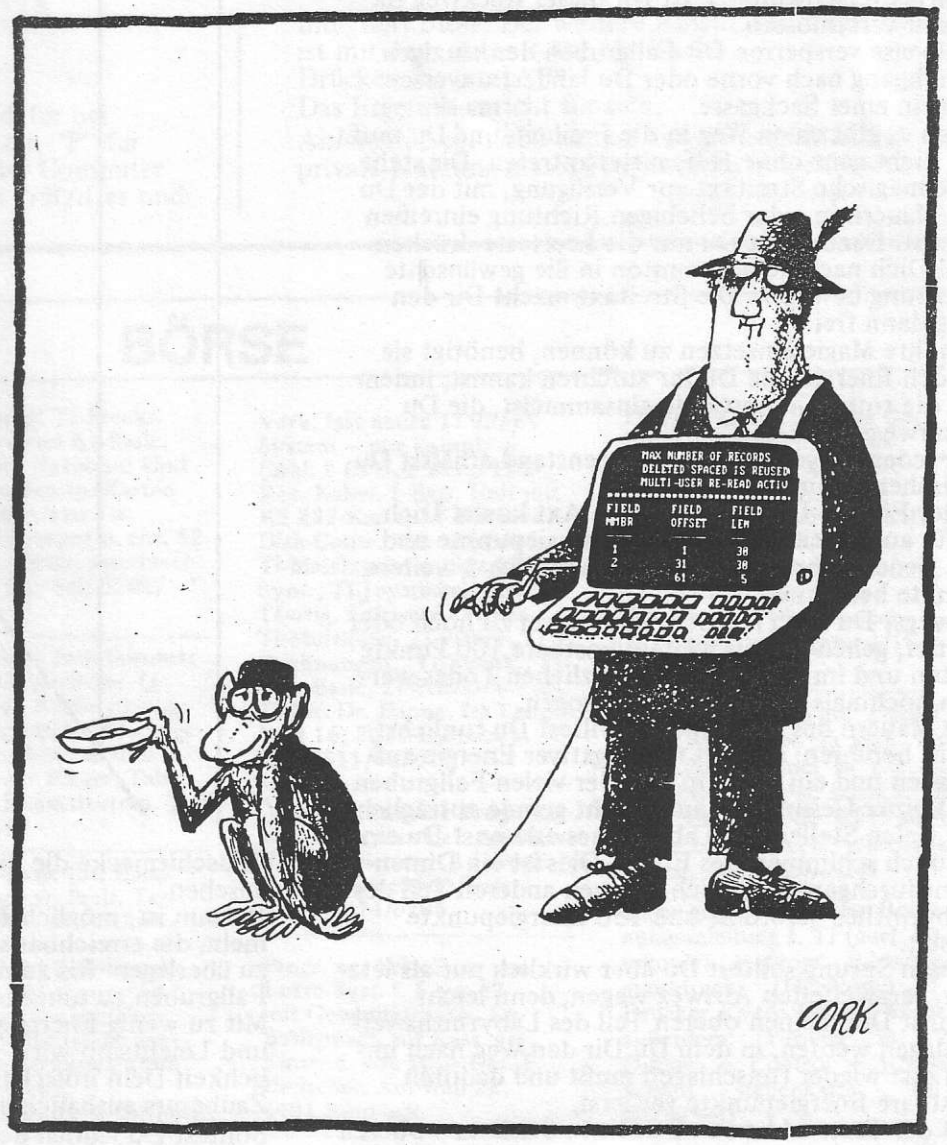
Haushaltsrechnung
Der TI99 hat Ihre Kasse fest im Griff. Mit Tendenzanzeige und Jahresgrafik — ab Seite 90

Rechtskosten
Ein Zivilprozeß kann teuer werden — ab Seite 97

Speech Prozessor 2
Der Sprachsynthesizer kann viel mehr, als nur englisch sprechen — ab Seite 102

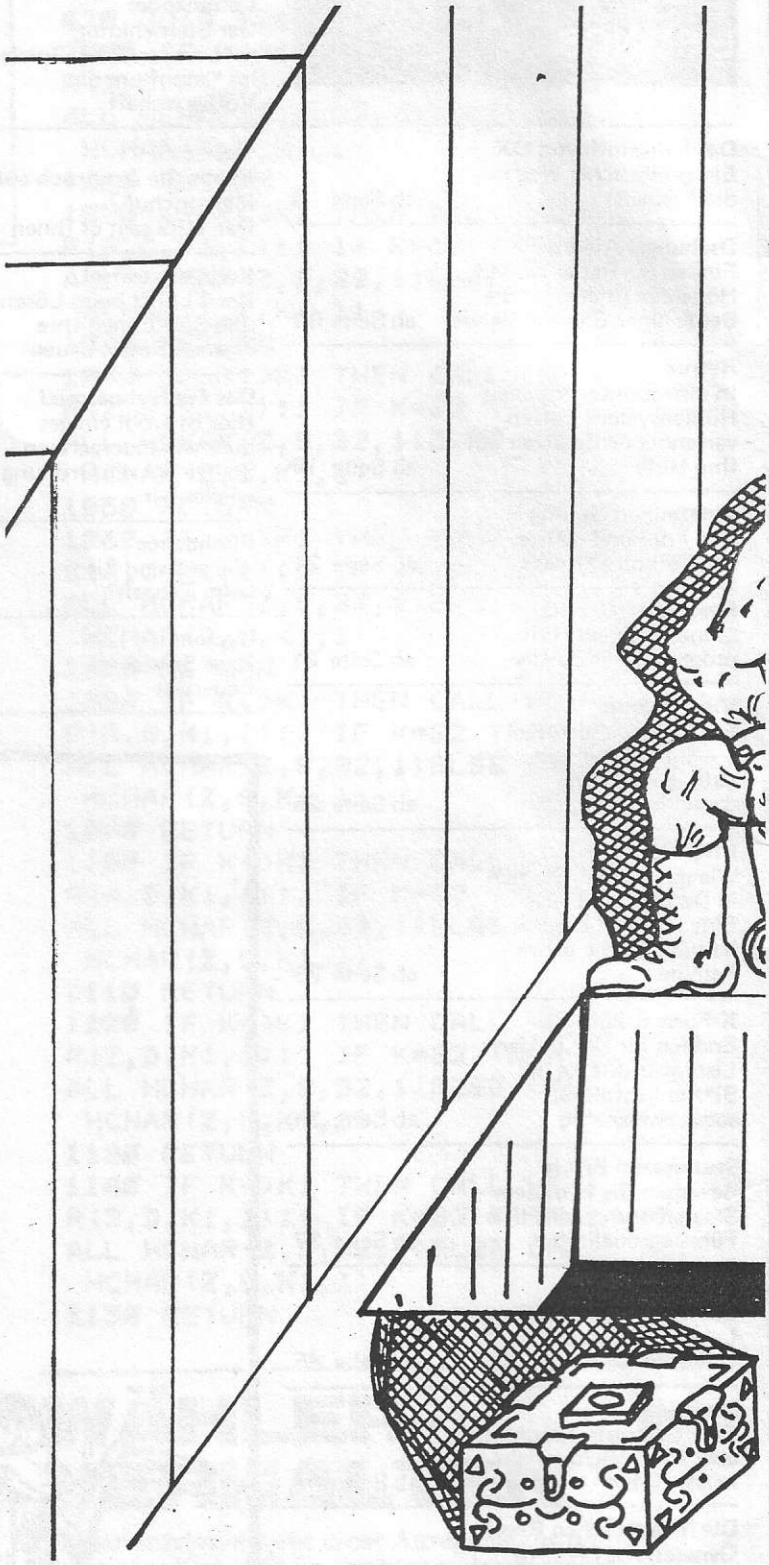
Das Berg'sche Spiel
Der TI gewinnt (fast) immer. Durchschauen Sie das System — ab Seite 110

EAN-Coder 13
Strichcodes erstellen prüfen, ausdrucken. Die erstellten Codes können von jeder Warenkasse gelesen werden — ab Seite 114

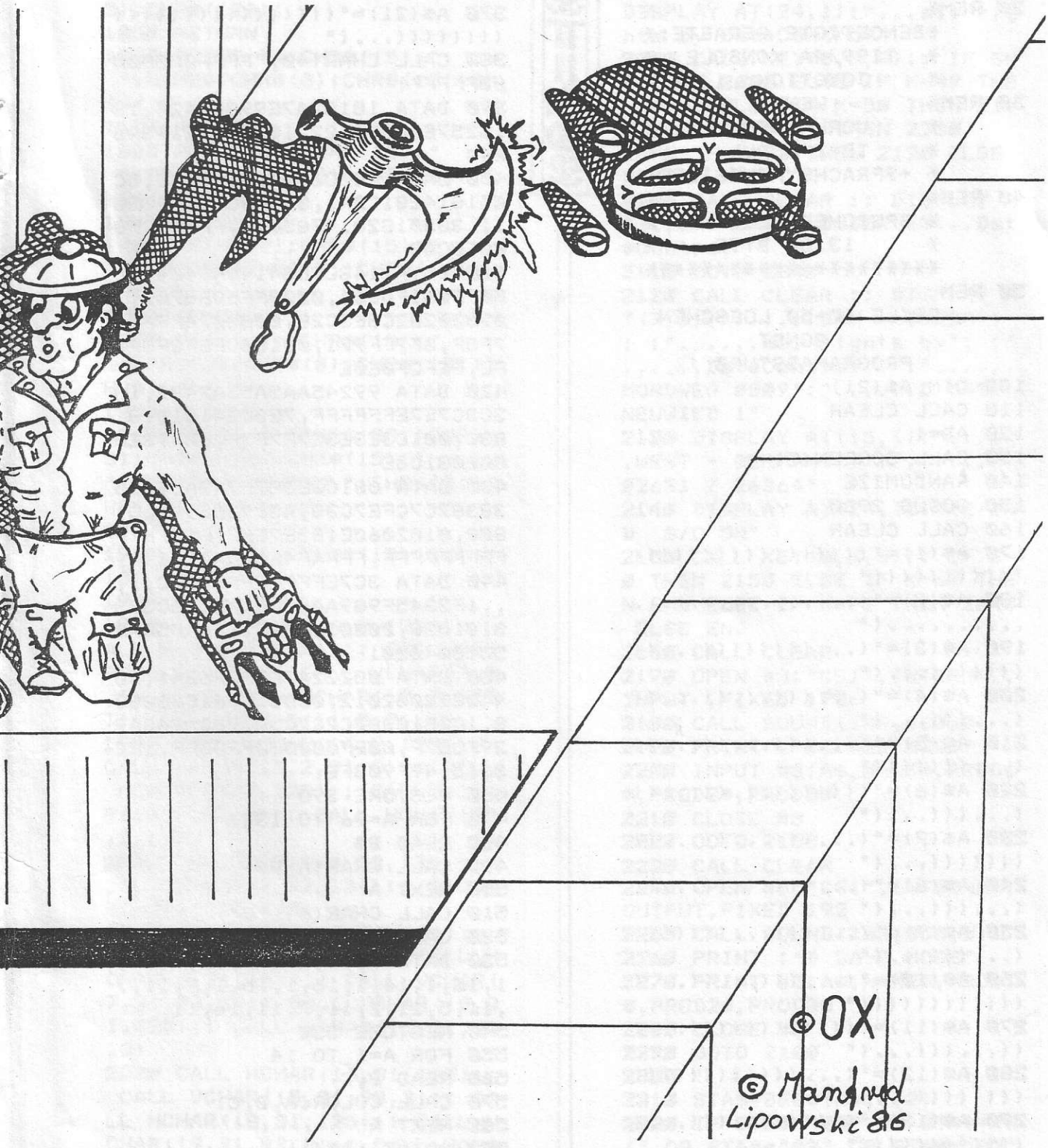


DAS LABYRINT VON OX

Der Zauberer Galmot hat Dich in das Labyrinth von Ox verzaubert.
Fast über 18 Bildschirme erstreckt sich das Labyrinth, in dem Du verzweifelt nach einem Ausgang suchst.
Erst wenn Du alle Gefahren überwunden hast, siehst Du in der Ferne den weißen Ausgang aus der Welt des Zauberers.
Bis dahin ist aber noch ein weiter Weg, den nur der beste und intelligenteste Held lebend meistern kann, wobei er ständig sein einziges Leben aufs Spiel setzen muß.
Du kannst Dich in den oberen zwei Dritteln des Bildschirms fast gefahrlos bewegen, willst Du tiefer in das Labyrinth eindringen, drückst Du einfach den Feuerknopf des Joysticks.
Es gibt jedoch keinen Weg zurück. Sind erst einmal Teile des Labyrints aus dem oberen Teil des Bildschirms verschwunden, ist Dir dieser Rückweg für immer verschlossen.
Teilweise versperren Dir Fallgruben den einzigen Durchgang nach vorne oder Du landest unversehens in einer Sackgasse.
Doch es gibt einen Weg in die Freiheit, und Du mußt ihn nicht ganz ohne Hilfsmittel antreten. Dir steht eine magische Streitaxt zur Verfügung, mit der Du die Mauern in jeder beliebigen Richtung einreißen kannst. Dazu mußt Du nur die Leertaste drücken und Dich nach dem Summton in die gewünschte Richtung bewegen. Die Streitaxt macht Dir den Weg dann frei.
Um ihre Magie einsetzen zu können, benötigt sie jedoch Energie, die Du ihr zuführen kannst, indem Du die roten Gegenstände einsammelst, die Du unterwegs findest.
Für jeden aufgenommenen Gegenstand erhältst Du zwischen 50 und 300 Punkte.
Jeder Einsatz Deiner magischen Axt kostet Dich dafür auch gleich wieder 100 Energiepunkte und für jeden Schritt werden Dir zusätzlich 5 weitere Punkte berechnet.
Bewegst Du Dich auf dem Bildschirm zu hoch oder zu tief, gehen jeweils weitere kostbare 100 Punkte flöten und im Falle Deines plötzlichen Todes werden nochmals 250 Punkte abgezogen.
Die Mauern des Labyrinthes solltest Du tunlichst nicht berühren; sie sind mit negativer Energie aufgeladen und ein Sturz in eine der vielen Fallgruben ist Deiner Gesundheit auch nicht gerade zuträglich.
An vielen Stellen des Labyrinthes erkennst Du ein bläulich schimmerndes Etwas. Dies ist ein Dimensionsdurchgang, der Dich in einen anderen Teil des Labyrinthes befordert und 150 Energiepunkte kostet.
Diesen Sprung solltest Du aber wirklich nur als letzten, verzweifelten Ausweg wagen, denn leicht kannst Du in einen oberen Teil des Labyrinths verschlagen werden, in dem Du Dir den Weg nach unten erst wieder freischlagen mußt und dadurch kostbare Energiepunkte verlierst.
Mit der Taste 's' kannst Du Dir in der linken oberen Bildschirmecke die Summe Deiner Energiepunkte ansehen.
Ratsam ist, möglichst alle Gegenstände aufzusammeln, die erreichbar sind und jeden Schritt genau zu überlegen. Bis zum rettenden Ausgang sind viele Fallgruben zu umgehen und Mauern einzureißen.
Mit zu wenig Energiepunkten oder zu viel Übermut und Leichtsinn wirst Du mit größter Wahrscheinlichkeit Dein irdisches Leben im Labyrinth des Zauberers aushauchen.
Solltest Du jedoch den Ausgang lebend errei-



LISTING



chen, wartet eine kleine Überraschung auf Dich. Obwohl dieses Spiel für TI-Basic und TE-II-Modul geschrieben ist, läuft es aber auch ohne Probleme und Änderungen in Extended-Basic.
Die Punkte im Listing gehören nicht zum Bildschirmaufbau. Sie sollen nur, wie immer, das Abtippen der vielen Leerzeichen erleichtern.
Zum Schluß noch einmal eine Übersicht über die Steuerung und die Punkteverteilung:

- Joystick (1) = Bewegung in alle Richtungen, einschließlich d iagonal (spart Energie) möglich.
- Feuerknopf = Labyrinth scrollt nach oben (nicht zurücknehmbar)
- Taste 's' = Score wird links oben angezeigt.
- Spacetaste = Gebrauch der Streitaxt

Punkte:

- Gegenstand aufnehmen = + 50 – + 300 Punkte
- 1 Schritt = – 5 Punkte
- Dimensionstor = –150 Punkte
- Streitaxteinsatz = –100 Punkte
- zu hoch/tief laufen = –100 Punkte
- Todesfall = –250 Punkte

LISTINGS

```
10 REM**********************
      *          OX          *
      * MANFRED LIPOWSKI     *
      *      (c)  1986       *
20 REM*                      *
      *BENOETIGTE GERAETE:*
      *  TI99/4A KONSOLE     *
      *    JOYSTICK  1       *
30 REM*        WENN          *
      *     VORHANDEN        *
      *     TE-2 MODUL       *
      * +SPRACHSYNTHESIZER*
40 REM*                      *
      * SPEICHERBELEGUNG:    *
      *      13104 BYTE      *
      **********************
50 REM
      ZEILE 10-50 LOESCHEN
            SONST
        PROGRAMABSTURZ !
100 DIM A$(21)
110 CALL CLEAR
120 AB=1
130 CALL SCREEN(5)
140 RANDOMIZE
150 GOSUB 2980
160 CALL CLEAR
170 A$(1)="((((((((((((((((((((
((((((((((("
180 A$(2)="(.................
..........("
190 A$(3)="(...(((((((((...((
(((((((...("
200 A$(4)="(...(((...(((...((
(...(((...("
210 A$(5)="(...(((...(((...((
(...(((((((("
220 A$(6)="(((((((...(((...((
(...(((...("
230 A$(7)="(...(((...(((...((
(((((((...("
240 A$(8)="(...(((((((((...((
(...(((...("
250 A$(9)="(...(((...(((((((
(...(((...("
260 A$(10)="(...(((...(((...(
((((((((((("
270 A$(11)="((((((((((((...(
((...(((...("
280 A$(12)="(...((((((((((((
(((((((...("
290 A$(13)="((((((((((((...(
((((((((((("
300 A$(14)="(..............(
((((((((((((("
310 A$(15)="((((((((((((....
..........("
320 A$(16)="( ( ( ( ( ( ( (
( ( ( ( ( (("
330 A$(17)="(( ( ( ( ( ( ( (
 ( ( ( ( ( ("
340 A$(18)="(...((((((((((((
((((((((((((("
350 A$(19)="(((((((...((((((
((((((((((("
360 A$(20)="((((((((((((((((
((...(((((((("
370 A$(21)="((((((((((((((((
(((((((...("
380 CALL CHAR(40,"FFFF000000
00FFFF")
390 DATA 1818247E993C2424,18
18257E985C1622,1818A47E193A6
844
400 DATA 0C0C317E581C6488,0C
0C1814281C0A1,3030BC7E1A3826
11,3030182814385008,FFFEFCF8
F0E0C08
410 DATA 143C7C3414040404,00
00FF061F0E04,0000FF60F8702,2
020202C3E3C28,0003074FFFFF
7F3F,3F7FF971,0018BCFEFCF8FC
FE,FEFCF0E0E
420 DATA 99245AA5A55A2499,18
3C3C7E7EFFFFFF,7E3C241818181
83C,081C3E3E3E7F7F7F,6B493E0
808081C3E
430 DATA 081C3E3E7F7F2A08,10
38387C7CFE7C38,48390E380B6C1
808,0102060E1E3E7EFF,FFFFFFF
FFFFFFFFF,FFFAF4E8D0A0C08
440 DATA 3C7EFFFFFFFF7E3C,,,
,,1F2345F989AA8CF8,00006C383
810103B,0000140814204,A050A0
5008040201
450 DATA 00222A7F7F7F6341,00
44082208201 2,00081C361C08080
8,1038103B7C7C7C38,78C3868A9
29FC07F,00FF000000FF00FF,1EC
3615149F903FE
460 RESTORE 390
470 FOR A=96 TO 138
480 READ B$
490 CALL CHAR(A,B$)
500 NEXT A
510 CALL CHAR(64,"0")
520 CALL SCREEN(5)
530 DATA 5,11,16,13,16,1,16,
1,16,1,16,1,16,1,16,1,2,11,7
,11,5,11,2,11,7,11,16,11
540 RESTORE 530
550 FOR A=1 TO 14
560 READ B,C
570 CALL COLOR(A,B,C)
580 NEXT A
590 PRINT :A$(1)
600 FOR A=1 TO 22
610 PRINT A$(2)
620 NEXT A
630 FOR A=1 TO 3
640 B=INT(RND*18)+3
650 C=INT(RND*23)+4
660 CALL HCHAR(B,C,INT(RND*8
)+128)
670 CALL SOUND(-99,-A,A)
680 NEXT A
```

LISTINGS

```
690 GOSUB 3220
700 D$="^FIND ^>THE _WAY ^>O
UT . ^>OX"
710 GOSUB 3320
720 SC=0
730 XX=5
740 YY=16
750 CALL HCHAR(XX,YY,96)
760 PA=96
770 D$="^THIS ^>ARE ^YOUR .
^ADVENTURE ^>MANN"
780 GOSUB 3320
790 RESTORE 2850
800 IF SC<0 THEN 810 ELSE 82
0
810 SC=0
820 CALL HCHAR(XX,YY,96)
830 CALL JOYST(1,X,Y)
840 CALL KEY(1,K,S)
850 IF K=18 THEN 1760
860 CALL KEY(0,M,L)
870 IF L=0 THEN 890
880 GOTO 2220
890 IF (X=0)*(Y=0)THEN 800
900 SC=SC-5
910 IF X=-4 THEN 950
920 IF X=4 THEN 970
930 PA=97
940 GOTO 980
950 PA=101
960 GOTO 980
970 PA=99
980 CALL HCHAR(XX,YY,32)
990 XX=XX+(Y/-4)
1000 YY=YY+(X/4)
1010 CALL GCHAR(XX,YY,L)
1020 IF L<>32 THEN 1280
1030 IF XX>1 THEN 1120
1040 XX=XX+1
1050 FOR A=0 TO 30 STEP 5
1060 CALL SOUND(-333,110+A*1
0,A,112+A*10,A)
1070 NEXT A
1080 D$="^>GO . ^>DOWN"
1090 SC=SC-100
1100 GOSUB 3320
1110 GOTO 3550
1120 IF XX<=20 THEN 1230
1130 XX=XX-1
1140 CALL HCHAR(XX,YY,96)
1150 D$="^>GO ^>HIGHER"
1160 FOR A=1 TO 3
1170 CALL SOUND(250,311*A,A,
415*A,A)
1180 CALL SOUND(425,208*A,A,
277*A,A)
1190 NEXT A
1200 GOSUB 3320
1210 SC=SC-100
1220 GOTO 3550
1230 CALL HCHAR(XX,YY,PA)
1240 CALL SOUND(-9,-5,9)
1250 CALL HCHAR(XX,YY,PA+1)
1260 CALL SOUND(-9,-5,15)
1270 GOTO 830
1280 CALL HCHAR(XX,YY,32)
1290 IF XX<=1 THEN 1040
1300 IF XX>=21 THEN 1130
1310 CALL HCHAR(XX,YY,96)
1320 IF L=112 THEN 2700
1330 IF (L=120)+(L=121)+(L=1
22)THEN 1420
1340 IF L=40 THEN 1640
1350 IF (L=136)+(L=137)+(L=1
38)THEN 3720
1360 SC=SC+INT(RND*250)+50
1370 B=INT(RND*900)+220
1380 FOR A=1 TO 30 STEP 7
1390 CALL SOUND(-99,B,A,B+A,
A,B*A/2,A)
1400 NEXT A
1410 GOTO 800
1420 CALL CLEAR
1430 CALL COLOR(1,2,2)
1440 CALL COLOR(10,11,11)
1450 CALL SCREEN(7)
1460 CALL COLOR(9,16,2)
1470 CALL COLOR(2,5,5)
1480 CALL VCHAR(1,23,40,480)
1490 CALL HCHAR(24,11,104,12
)
1500 FOR A=1 TO 23
1510 B=INT(RND*7)+96
1520 CALL HCHAR(A,16,B)
1530 CALL HCHAR(A,16,32)
1540 CALL SOUND(-222,900-A*1
0,A,800-A*11,A)
1550 NEXT A
1560 FOR A=0 TO 30 STEP 2.5
1570 CALL SOUND(-99,-5,A)
1580 NEXT A
1590 CALL CLEAR
1600 CALL SCREEN(2)
1610 SC=SC-250
1620 AB=2
1630 GOTO 2980
1640 FOR A=1 TO 30 STEP 2
1650 CALL HCHAR(XX,YY,96)
1660 CALL COLOR(2,13,16)
1670 CALL HCHAR(XX,YY,40)
1680 CALL COLOR(2,16,13)
1690 CALL SOUND(-333,-6,A)
1700 NEXT A
1710 SC=SC-250
1720 CALL SCREEN(14)
1730 CALL CLEAR
1740 AB=2
1750 GOTO 2980
1760 READ KO
1770 IF KO=23 THEN 1780 ELSE
 1800
1780 GOSUB 3630
1790 KO=4
1800 CALL HCHAR(XX,YY,32)
1810 IF KO=22 THEN 1820 ELSE
 1840
```

```
1820 PRINT A$(INT(RND*20)+1)
1830 GOTO 1850
1840 PRINT A$(KO)
1850 IF RND<.4 THEN 1870
1860 GOTO 1010
1870 A=INT(RND*3)+1
1880 ON A GOTO 1890,1990,259
0
1890 FOR A=1 TO 4
1900 B=INT(RND*23)+4
1910 CALL GCHAR(22,B,C)
1920 IF C<>32 THEN 1970
1930 CALL HCHAR(22,B,112)
1940 A=INT(RND*900)+220
1950 CALL SOUND(-333,A,30,A,
30,A,30,-4,0)
1960 GOTO 1980
1970 NEXT A
1980 GOTO 1010
1990 IF (KO=1)+(KO=16)+(KO=1
7)THEN 1010
2000 B=INT(RND*5)+1
2010 ON B GOTO 2020,2040,206
0,2080,2100
2020 B=4
2030 GOTO 2110
2040 B=10
2050 GOTO 2110
2060 B=15
2070 GOTO 2110
2080 B=21
2090 GOTO 2110
2100 B=27
2110 CALL GCHAR(22,B,C)
2120 CALL GCHAR(22,B+1,C1)
2130 CALL GCHAR(22,B+2,C2)
2140 IF (C<>32)+(C1<>32)+(C2
<>32)THEN 1010
2150 CALL HCHAR(22,B,120)
2160 CALL HCHAR(22,B+1,121)
2170 CALL HCHAR(22,B+2,122)
2180 FOR A=2 TO 26 STEP 8
2190 CALL SOUND(-99,220-A*3,
A,-5,A+4)
2200 NEXT A
2210 GOTO 1010
2220 IF M=115 THEN 3550
2230 IF SC<=99 THEN 800
2240 CALL JOYST(1,X,Y)
2250 CALL SOUND(1,-1,20)
2260 IF (X=0)*(Y=0)THEN 2240
2270 SC=SC-100
2280 IF X=-4 THEN 2290 ELSE
2310
2290 P1=106
2300 GOTO 2410
2310 IF X=4 THEN 2320 ELSE 2
340
2320 P1=105
2330 GOTO 2410
2340 IF Y=4 THEN 2350 ELSE 2
370
2350 P1=104
2360 GOTO 2410
2370 IF Y=-4 THEN 2380 ELSE
2400
2380 P1=107
2390 GOTO 2410
2400 GOTO 800
2410 X1=XX+(Y/-4)
2420 Y1=YY+(X/4)
2430 CALL GCHAR(X1,Y1,I)
2440 IF I<>40 THEN 2450 ELSE
 2510
2450 IF (I=120)+(I=121)+(I=1
22)THEN 2460 ELSE 2490
2460 SC=SC-250
2470 CALL SOUND(-99,-7,0)
2480 GOTO 800
2490 SC=SC-100
2500 GOTO 800
2510 FOR A=1 TO 5
2520 CALL HCHAR(X1,Y1,I)
2530 CALL SOUND(-99,-7,A*2)
2540 CALL HCHAR(X1,Y1,P1)
2550 CALL SOUND(-99,-6,A*2)
2560 NEXT A
2570 CALL HCHAR(X1,Y1,32)
2580 GOTO 800
2590 Q=INT(RND*8)+128
2600 FOR A=1 TO 4
2610 C=INT(RND*24)+3
2620 CALL GCHAR(22,C,D)
2630 IF D<>32 THEN 2680
2640 CALL HCHAR(22,C,Q)
2650 A=INT(RND*900)+1500
2660 CALL SOUND(-99,A,30,A,3
0,A,30,-8,0)
2670 GOTO 1010
2680 NEXT A
2690 GOTO 1010
2700 CALL HCHAR(XX,YY,32)
2710 IF SC<=149 THEN 800
2720 CALL SOUND(-99,-7,0)
2730 SC=SC-150
2740 FOR A=900 TO 500 STEP -
25
2750 CALL SOUND(-333,A,9,A+A
,15)
2760 XX=INT(RND*18)+3
2770 YY=INT(RND*23)+4
2780 CALL GCHAR(XX,YY,D)
2790 IF D<>32 THEN 2830
2800 CALL HCHAR(XX,YY,96)
2810 CALL SOUND(-99,-6,0)
2820 GOTO 800
2830 NEXT A
2840 GOTO 2740
2850 DATA 4,4,4,4,4,4,4,4,4,
4,4,4,6,6,6,6,6,6,6,6,5,5,5,
5,5,9,9,9,9,9,2,2,2,2,2,2,
2,2
2860 DATA 12,12,12,12,12,12,
12,12,12,12,12,12,12,12,14,14,14,1
4,14,14,15,15,15,15,15
2870 DATA 2,2,2,2,2,2,2,2,4,
```

LISTINGS

```
4,4,4,4,4,4,4,4,4,4,4,4,4,4,
9,9,9,9,9,9,9,9,10,10,10,10,
10
2880 DATA 14,14,14,14,16,16,
17,17,16,16,17,17,16,16,17,1
7,21,21,21,21,21,21,21,21,21
,21
2890 DATA 3,3,3,3,3,3,3,3,3,
4,4,4,4,12,12,2,2,18,18,18,1
8,2,2,2,2,21,21,21,21,16,17,
16,17
2900 DATA 16,17,16,17,16,17,
18,18,18,18,18,18,18,18,2,2,
2,2,2,4,4,4,4,4,4,4,4,4,4,4,
4,4,6,6,6,6
2910 DATA 9,9,9,9,5,5,5,5,5,
14,14,14,14,14,16,17,21,21,2
1,21,21,21,16,17,18,18,18,18
,18,2,2,2,2
2920 DATA 3,3,3,3,3,3,3,3,3,
3,4,4,4,4,4,4,4,4,4,4,5,5,5,
5,5,14,14,14,14,10,10,10,10,
10,10
2930 DATA 10,10,22,22,22,16,
17,16,11,11,11,11,11,11,11,2
2,22,22,22,11,11,11,7,7,7,7,
21,21,21,21
2940 DATA 22,21,21,2,2,18,18
,18,2,21,21,21,17,16,17,17,1
6,16,17,18,18,18,18,18,4,4,4
,4,4,4,4,4
2950 DATA 22,22,22,12,12,12,
12,12,12,12,12,12,12,18,18,1
8,18,18,16,17,16,17,16,17,16
,17,16,17
2960 DATA 16,17,16,17,22,22,
22,22,3,3,3,3,3,3,3,3,9,9,9,
9,9,9,12,12,12,21,21,21
2970 DATA 22,22,22,21,21,16,
17,18,18,18,2,2,2,2,23,2,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1
2980 FOR A=1 TO 14
2990 CALL COLOR(A,2,11)
3000 NEXT A
3010 CALL CHAR(35,"3C4299A1A
199423C")
3020 PRINT : : : : : : : :"
 MANFRED LIPOWSKI PRESENT":
: : : :".............OX": :
: : :
3030 PRINT " # APRIL 1986 BY
 M.LIPOWSKI": : : : : :
3040 IF AB<>1 THEN 3400
3050 PRINT :".....SIND TE-2
 MODUL +": :".....SPRACHSYN
THESIZER": :".......ANGESCHL
OSSEN ?"
3060 CALL KEY(0,A,B)
3070 IF B=0 THEN 3060
3080 IF (A<>74)*(A<>106)THEN
 3130
3090 OPEN #1:"SPEECH",OUTPUT
```

LISTINGS

```
3100 PRINT #1:">^WELCOME ^TO
O . ^OX"
3110 Z=1
3120 GOTO 3140
3130 Z=0
3140 RETURN
3150 GOTO 3150
3160 D=INT(RND*999)+600
3170 FOR C=1 TO LEN(B$)
3180 CALL HCHAR(A,B+C,ASC(SE
G$(B$,C,1)))
3190 CALL SOUND(-99,D,30,D,3
0,D,30,-4,C)
3200 NEXT C
3210 RETURN
3220 DATA 294,262,233,220,19
6,196,196,30000,294,262,233,
220,196,196,196,30000
3230 DATA 392,349,311,294,26
2,262,262,30000,294,262,233,
220,196,196,196,30000
3240 RESTORE 3220
3250 FOR A=1 TO 32
3260 READ B
3270 FOR C=0 TO 30 STEP 15
3280 CALL SOUND(-99,B,C,B-2,
C)
3290 NEXT C
3300 NEXT A
3310 RETURN
3320 IF Z<>1 THEN 3360 ELSE
3330
3330 CALL SOUND(1,110,30)
3340 PRINT #1:D$
3350 RETURN
3360 FOR A=30 TO 0 STEP -5
3370 CALL SOUND(-99,-5,A)
3380 NEXT A
3390 RETURN
3400 IF SC<0 THEN 3410 ELSE
3420
3410 SC=0
3420 IF AB<>3 THEN 3450
3430 PRINT :" DU HAST ES GES
CHAFFT"
3440 GOTO 3460
3450 PRINT :" DU HAST ES NIC
HT GESCHAFFT"
3460 PRINT :" DEIN PUNKTESTA
ND";SC;"": :" NOCHMAL ? J/N"
3470 D$="^THIS ^IS ^>THE ^>E
ND"
3480 GOSUB 3320
3490 CALL KEY(0,A,B)
3500 IF B=0 THEN 3490
3510 IF (A<>74)*(A<>106)THEN
 3530
3520 GOTO 160
3530 CALL CLEAR
3540 END
3550 CALL VCHAR(1,2,64,24)
3560 IF SC<0 THEN 3570 ELSE
3580
```

```
3570 SC=0
3580 FOR A=1 TO LEN(STR$(SC)
)
3590 CALL VCHAR(1+A,2,ASC(SE
G$(STR$(SC),A,1)))
3600 CALL SOUND(1,999,9)
3610 NEXT A
3620 GOTO 800
3630 CALL HCHAR(22,15,136)
3640 CALL HCHAR(22,16,137)
3650 CALL HCHAR(22,17,138)
3660 FOR A=0 TO 30 STEP 5
3670 CALL SOUND(-99,220,A,19
6,A)
3680 CALL SOUND(-99,196,A,33
0,A)
3690 CALL SOUND(-99,330,A,22
0,A)
3700 NEXT A
3710 RETURN
3720 D$="^>YOU ^FIND ^>THE _
>WAY . ^OUT ^>OX"
3730 GOSUB 3320
3740 CALL CLEAR
3750 CALL SCREEN(11)
3760 FOR A=1 TO 14
3770 CALL COLOR(A,1,1)
3780 NEXT A
3790 CALL CHAR(128,"0103070F
1F3F7FFF")
3800 CALL CHAR(136,"FFFEFCF8
F0E0C080")
3810 CALL CHAR(129,"FFFFFFFF
FFFFFFFF")
3820 CALL CHAR(130,"FFFEFCF8
F0E0C080")
3830 CALL CHAR(137,"0")
3840 FOR A=1 TO 21
3850 A$(A)=""
3860 NEXT A
3870 CALL HCHAR(20,5,128)
3880 CALL HCHAR(21,4,128)
3890 CALL HCHAR(21,5,136)
3900 CALL HCHAR(20,6,129,2)
3910 CALL HCHAR(20,8,130)
3920 CALL HCHAR(21,6,137)
3930 CALL HCHAR(21,7,103)
3940 CALL HCHAR(1,1,40,256)
3950 CALL HCHAR(3,16,123)
3960 FOR A=1 TO 31 STEP 2
3970 CALL HCHAR(17,A,113)
3980 CALL HCHAR(18,A,114)
3990 CALL HCHAR(17,A+1,115)
4000 CALL HCHAR(18,A+1,116)
4010 NEXT A
4020 A=1
4030 B=4
4040 GOSUB 4440
4050 A=3
4060 B=27
4070 GOSUB 4440
4080 A=4
4090 B=13
```

LISTINGS

```
4100 GOSUB 4440
4110 FOR A=1 TO 20
4120 CALL HCHAR(INT(RND*7)+9
,INT(RND*31)+1,INT(RND*3)+11
7)
4130 NEXT A
4140 DATA 11,11,5,5,16,1,16,
1,2,1,2,1,2,1,2,1,2,1,16,5,1
3,1,12,5,7,1,7,2
4150 RESTORE 4140
4160 FOR A=1 TO 14
4170 READ B,C
4180 CALL COLOR(A,B,C)
4190 NEXT A
4200 FOR A=1 TO 8
4210 FOR B=24 TO 19 STEP -1
4220 CALL HCHAR(B,11,INT(RND
*7)+96)
4230 FOR C=1 TO 20
4240 NEXT C
4250 CALL HCHAR(B,11,32)
4260 CALL SOUND(-333,600-B*1
0,8)
4270 NEXT B
4280 FOR B=19 TO 24
4290 CALL HCHAR(B,11,INT(RND
*7)+96)
4300 FOR C=1 TO 20
4310 NEXT C
4320 CALL HCHAR(B,11,32)
4330 CALL SOUND(-333,600+B*1
0,8)
4340 NEXT B
4350 NEXT A
4360 CALL HCHAR(24,11,96)
4370 AB=3
4380 FOR A=1 TO 999
4390 NEXT A
4400 CALL CLEAR
4410 CALL SCREEN(13)
4420 GOSUB 2980
4430 GOTO 100
4440 CALL HCHAR(A,B,108)
4450 CALL HCHAR(A+1,B,109)
4460 CALL HCHAR(A,B+1,110)
4470 CALL HCHAR(A+1,B+1,111)
4480 RETURN
```

JUNGLE-ADVENTURE

Seit Monaten schon trieb eine Gangsterbande in allen Großstädten ihr Unwesen und raubte alles, was nicht niet- und nagelfest war. Als die Bande nun letzte Woche gefangen wurde, blieb die Beute verschwunden, und nichts konnte sie dazu bewegen, zu verraten, wo die Beute versteckt ist. Erst nach stundenlangen Verhören und der Zusicherung auf Straffreiheit konnte einem Mitglied ein

Bitte lesen Sie weiter auf Seite 14

LISTINGS

```
10 ! **********************
11 ! *                    *
12 ! * Dschungel-Abenteuer*
13 ! *                    *
14 ! *    Copyright by    *
15 ! *  Matthias Renken   *
16 ! *                    *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     9150 Bytes     *
28 ! *                    *
29 ! **********************Pr
     ogramm bitte mit abtipp
     en.
100 CALL CLEAR :: FOR F=65 T
O 90 :: CALL CHARPAT(F,E$)::
 CALL CHAR(F+32,E$):: NEXT F
110 CALL CHAR(121,"00FF00000
0000000",120,"0020100804081
020")
120 CALL CHAR(123,"0000303000
0303000",125,"0038440408100
010")
130 CALL CHAR(64,"3C4299A1A1
99423C")
140 FOR F=1 TO 14 :: CALL CO
LOR(F,8,5):: NEXT F :: CALL
SCREEN(2)
150 DISPLAY AT(5,7):"JUNGLE-
ADVENTURE" :: DISPLAY AT(8,1
4):"BY"
160 DISPLAY AT(11,8):"MATTHI
AS RENKEN" :: DISPLAY AT(18,
3):"@ AUG. 1985"
170 CALL KEY(0,K,S):: IF S=0
 THEN 170
180 CALL CLEAR
190 CALL COLOR(0,8,5,4,8,5)
200 CALL COLOR(9,6,5,10,6,5)
210 CALL COLOR(11,6,5,12,6,5
)
220 ORT=11 :: DIM BEF$(11),G
EG$(25,3),UNT$(25),UMG$(25),
FELD(25,4),INV$(5)
230 INSCH$="'DIE VERSPERRTEN
 WOGEN DES   FEUCHTEN FUEHRE
N ZUM        HIMMELREICH'"
240 RANDOMIZE :: FOR E=1 TO
5 :: CO$=CO$&CHR$(INT(RND*25
)+65):: NEXT E
250 RESTORE :: FOR F=1 TO 11
 :: READ BEF$(F):: NEXT F
260 FOR F=1 TO 25 :: READ UM
G$(F):: NEXT F
270 UNT$(5)="BUSCHMESSER" ::
 GEG$(8,1)="SCHLUESSEL"
280 GEG$(9,1)="HOHES GRAS" :
: GEG$(11,1)="SCHLUESSEL"
290 GEG$(13,1)="KISTE" :: GE
G$(14,1)="STANDBILDER"
300 GEG$(18,1)="DICKICHT" ::
 GEG$(20,1)="EIMER"
310 GEG$(23,1)="HOEHLE" :: G
EG$(25,1)="STEIN"
320 GEG$(10,1)="DICKICHT" ::
 GEG$(15,1)="SCHLANGE" :: GE
G$(6,1)="GIFTIGE TIERE"
330 UNT$(9)="BRECHSTANGE" ::
 UNT$(14)="EINE INSCHRIFT "
340 UNT$(20)="ER IST VOLL WA
SSER" :: UNT$(23)="EIN MODER
NES ZAHLENSCHLOSS"
350 UNT$(25)="SCHLANGENFALLE
" :: UNT$(10)="EINE HOEHLE M
IT SCHLOSS"
360 FOR E=1 TO 25 :: IF GEG$
(E,1)="" THEN GEG$(E,1)="KEI
NE"
370 NEXT E
380 FOR T=1 TO 25
390 FOR Y=1 TO 4
400 READ FELD(T,Y)
410 NEXT Y
420 NEXT T
430 !
440 CALL CLEAR :: PRINT :"ic
h sehe hier(                YY
YYYYYYYYYYY":UMG$(ORT): :
450 PRINT "besonderheiten(
              YYYYYYYYYYYYYYY":
460 FOR G=1 TO 3 :: IF GEG$(
ORT,G)<>"" THEN PRINT GEG$(O
RT,G)
470 NEXT G :: PRINT :"moegli
che richtungen(         YYYYYY
YYYYYYYYYYYYY":
480 IF FELD(ORT,1)<>0 THEN P
RINT "NORD"
490 IF FELD(ORT,2)<>0 THEN P
RINT "OST"
500 IF FELD(ORT,3)<>0 THEN P
RINT "SUED"
510 IF FELD(ORT,4)<>0 THEN P
RINT "WEST"
520 PRINT "YYYYYYYYYYYYYYYYY
YYYYYYYYYYY":
530 PRINT "was soll ich tun
)"
540 INPUT "x":IN$ :: PRINT
550 IF LEN(IN$)>1 THEN 580
560 IF IN$<>"N" AND IN$<>"S"
 AND IN$<>"O" AND IN$<>"W" T
HEN 580
570 NOM$=IN$ :: GOTO 900
580 PO=POS(IN$," ",1):: VERB
$=SEG$(IN$,1,3):: NOM$=SEG$(
IN$,PO+1,LEN(IN$)-PO)
590 FOR F=1 TO 11 :: IF VERB
$=BEF$(F)THEN 620
600 NEXT F
610 PRINT :"'";SEG$(IN$,1,PO
);"'":"VERSTEHE ICH NICHT!":
 :: GOTO 520
620 ON F GOTO 640,860,900,12
30,1320,1370,1320,1440,1580,
1620,430
```

LISTINGS

```
630 GOTO 520
640 IF NOM$="SCHLANGE" THEN
830
650 IF NOM$="TIERE" OR NOM$=
"GIFTIGE" THEN 840
660 IF NOM$="KISTE" THEN 850
670 IF GEG$(ORT,1)="" AND GE
G$(ORT,2)="" AND GEG$(ORT,3)
="" THEN PRINT "HIER GIBT ES
 NICHTS AUF-    ZUNEHMEN!" :
: GOTO 520
680 IF GEG$(ORT,1)<>NOM$ AND
 GEG$(ORT,2)<>NOM$ AND GEG$(
ORT,3)<>NOM$ THEN PRINT "DAS
 GIBT ES HIER NICHT !!" :: G
OTO 520
690 IF NOM$="BUSCHMESSER" TH
EN B=1 :: GOTO 770
700 IF NOM$="SCHLUESSEL" AND
 ORT=11 THEN KEY1=1 :: GOTO
770
710 IF NOM$="SCHLUESSEL" AND
 ORT=8 THEN KEY2=1 :: GOTO 7
70
720 IF NOM$="EIMER" THEN E=1
 :: GOTO 770
730 IF NOM$="BRECHSTANGE" TH
EN BR=1 :: GOTO 770
740 IF NOM$="SCHLANGENFALLE"
 THEN SCH=1 :: GOTO 770
750 IF NOM$="PERGAMENT" THEN
 PER=1 :: GOTO 770
760 PRINT "DAS KANN ICH NICH
T NEHMEN." :: GOTO 520
770 FOR T=1 TO 5 :: IF INV$(
T)<>"" THEN 780 ELSE INV$(T)
=NOM$ :: PRINT "ICH HABE ES.
" :: GOTO 800
780 NEXT T
790 PRINT "SO VIEL KANN ICH
NICHT      TRAGEN. ICH MUSS
ERST ETWAS WEGLEGEN!" :: GOT
O 520
800 FOR Y=1 TO 3 :: IF GEG$(
ORT,Y)<>NOM$ THEN 810 ELSE G
EG$(ORT,Y)="" :: GOTO 520
810 NEXT Y
820 GOTO 520
830 PRINT : :"AHHH, DIE SCHL
ANGE WAR      GIFTIG UND HAT
 MICH UMGE-  BRACHT !!!!" ::
 GOTO 1650
840 PRINT : :"DAS HAETTE ICH
 MIR AUCH DEN-KEN KOENNEN, D
ASS ICH DAS   HIER NICHT UEB
ERLEBE. DIE   TIERE HABEN MI
CH UMGEBRACHT." :: GOTO 1650
850 PRINT : :"VERFLU..., UNT
ER DER KISTE  WAREN GIFTIGE
DORNEN. JETZT MUSS ICH STERB
EN..." :: GOTO 1650
860 IF NOM$<>"WASSER" THEN P
RINT "KANNST DU MIR MAL SAGE
N, WIEMAN ";NOM$;" GIESST ?"
 :: GOTO 520
870 IF NOM$="WASSER" AND E=0
 THEN PRINT "DU HAST ES NICH
T." :: GOTO 520
880 IF ORT=6 AND E=1 THEN PR
INT "DIE TIERE FLUECHTEN VOR
 DEM WASSER DER WEG IST FREI
 !!!" :: TIEWEG=1 :: GOTO 52
0
890 PRINT "SCHOEN UND GUT, A
BER ES PAS-SIERT NICHTS! PEC
H." :: GOTO 520
900 HIL$=SEG$(NOM$,1,1)
910 IF HIL$<>"N" AND HIL$<>"
O" AND HIL$<>"S" AND HIL$<>"
W" THEN PRINT "DIESE RICHTUN
G GIBT ES NICHT" :: GOTO 540
920 IF HIL$="N" AND FELD(ORT
,1)=0 THEN 970
930 IF HIL$="O" AND FELD(ORT
,2)=0 THEN 970
940 IF HIL$="S" AND FELD(ORT
,3)=0 THEN 970
950 IF HIL$="W" AND FELD(ORT
,4)=0 THEN 970
960 GOTO 980
970 PRINT :"DORTHIN FUEHRT K
EIN WEG !!" :: GOTO 540
980 IF ORT=1 THEN 1040
990 IF ORT=22 THEN 1070
1000 IF ORT=6 THEN 1100
1010 IF ORT=15 THEN 1130
1020 IF ORT=18 THEN 1160
1030 GOTO 1180
1040 IF HIL$<>"O" THEN 1180
1050 IF BR=0 THEN PRINT "DA
LIEGEN DOCH  FELSEN!" :: GOT
O 520
1060 GOTO 1180
1070 IF HIL$<>"O" THEN 1180
1080 IF BO=0 THEN PRINT "ICH
 KANN NICHT SCHWIMMEN." :: G
OTO 520
1090 GOTO 1180
1100 IF HIL$<>"W" THEN 1180
1110 IF TIEWEG=0 THEN PRINT
"DA SIND DOCH DIE TIERE." ::
 GOTO 520
1120 GOTO 1180
1130 IF HIL$<>"N" THEN 1180
1140 IF SCH=0 THEN PRINT "DO
RT LIEGT EINE SCHLANGE!" ::
GOTO 520
1150 GOTO 1180
1160 IF HIL$<>"W" THEN 1180
1170 IF B=0 THEN PRINT "DURC
H DAS DICKICHT KOMME ICHNICH
T!" :: GOTO 520
1180 IF HIL$="N" THEN ORT=FE
LD(ORT,1)
1190 IF HIL$="O" THEN ORT=FE
LD(ORT,2)
1200 IF HIL$="S" THEN ORT=FE
```

LISTINGS

```
LD(ORT,3)
1210 IF HIL$="W" THEN ORT=FE
LD(ORT,4)
1220 GOTO 430
1230 IF UNT$(ORT)<>"" THEN P
RINT "ICH ENTDECKE:":UNT$(OR
T):: GOTO 1250
1240 PRINT UNT$(ORT):: GOTO
520
1250 IF ORT=14 OR ORT=20 OR
ORT=23 OR ORT=10 THEN 1290
1260 FOR Y=1 TO 3 :: IF GEG$
(ORT,Y)<>"" OR GEG$(ORT,Y)="
KEINE" THEN 1270 ELSE GEG$(O
RT,Y)=UNT$(ORT):: UNT$(ORT)=
"" :: GOTO 520
1270 NEXT Y
1280 PRINT "NIMM BITTE ETWAS
 WEG. UND   VERSUCHE ES NOCH
 EINMAL" :: GOTO 540
1290 IF ORT=10 THEN HOE=1 EL
SE IF ORT=14 THEN INSC=1
1300 IF ORT=23 THEN HOE2=1
1310 GOTO 520
1320 FOR R=1 TO 5 :: IF INV$
(R)<>NOM$ THEN 1330 ELSE 134
0
1330 NEXT R :: PRINT "DAS HA
BE ICH NICHT !!" :: GOTO 520
1340 FOR E=1 TO 3 :: IF GEG$
(ORT,E)="" THEN 1360 ELSE 13
50
1350 NEXT E :: PRINT "HIER L
IEGT SCHON SO VIEL.   NIMM E
RST ETWAS WEG." :: GOTO 520
1360 INV$(R)="" :: GEG$(ORT,
E)=NOM$ :: PRINT : :"GUT": :
: GOTO 520
1370 IF ORT<>14 AND ORT<>13
AND PER=0 THEN PRINT "HIER G
IBT ES NICHTS ZU LE-  SEN.":
 :: GOTO 520
1380 IF NOM$="PERGAMENT" THE
N PRINT CO$ :: GOTO 520
1390 IF ORT=13 THEN 1420
1400 IF INSC=0 THEN 1410 ELS
E PRINT INSCH$ :: GOTO 520
1410 PRINT "ICH KANN HIER NI
CHTS LESEN." :: GOTO 520
1420 IF KIS<>0 OR NOM$<>"PER
GAMENT" THEN 1410
1430 PRINT "AUF DEM ZETTEL S
TEHT":CO$ :: GOTO 520
1440 IF ORT<>10 AND ORT<>13
AND ORT<>23 THEN PRINT "HIER
 GIBT ES NICHTS ZU OEF- FNEN
." :: GOTO 520
1450 IF ORT=13 OR ORT=23 THE
N 1520
1460 IF HOE=0 THEN 1470 ELSE
 IF KEY2=1 THEN 1480 ELSE PR
INT "MIR FEHLT DER PASSENDE
     SCHLUESSEL." :: GOTO 52
0
1470 PRINT "ICH KANN HIER NI
CHTS OEF-   FNEN." :: GOTO 5
20
1480 PRINT "EINE HOEHLE OEFF
NET SICH    VOR MIR.ICH ENTD
ECKE EIN    BOOT IN IHR."
1490 BO=1
1500 PRINT "ICH NEHME ES SCH
ON EINMAL." :: FOR R=1 TO 5
:: IF INV$(R)<>"" THEN 1510
ELSE INV$(R)="BOOT" :: GOTO
520
1510 NEXT R :: PRINT "LEG ER
ST ETWAS WEG UND VER- SUCHE
DANN NOCH EINMAL       'OEFFN
E'" :: GOTO 520
1520 IF ORT=23 THEN 1560
1530 IF KEY2=0 THEN PRINT "I
CH HABE NICHT DEN          p
assenden SCHLUESSEL!!" :: GO
TO 520
1540 PRINT "DIE KISTE IST OF
FEN. IN IHR BEFINDET SICH EI
N PERGAMENT."
1550 GOTO 520
1560 IF HOE2=0 THEN PRINT "W
AS SOLL ICH DENN HIER OEF- F
NEN??" :: GOTO 520
1570 INPUT "ZUM OEFFNEN BRAU
CHE ICH DIE KOMBINATION DES
SCHLOSSES:":KOM$ :: IF KOM$=
CO$ THEN 2040 ELSE PRINT "NI
CHTS PASSIERT." :: GOTO 520
1580 PRINT : : :"WILLST DU W
IRKLICH AUF-     HOEREN ?"
1590 CALL KEY(0,K,S):: IF S=
0 THEN 1590
1600 IF K=74 THEN 1610 ELSE
430
1610 CALL CLEAR :: END
1620 PRINT : :"DU TRAEGST BE
I DIR:";
1630 FOR W=1 TO 5 :: PRINT I
NV$(W):: NEXT W
1640 PRINT : : :: GOTO 520
1650 REM ** TOT **
1660 PRINT : : : :"DIESER VE
RSUCH WAR JA NICHT SO ERFOLG
REICH... ABER PRO- BIERE ES
DOCH NOCH EINMAL!"
1670 CALL KEY(0,K,S):: IF S=
0 THEN 1670
1680 IF K=74 THEN 220 ELSE C
ALL CLEAR :: END
1690 REM ** BEFEHLE
1700 DATA NIM,GIE,GEH,UNT,WI
R,LIE,LEG,OEF,END,INV,SIE
1710 DATA "EIN HOHES GEBIRGE
 MIT        SCHNEEBEDECKTEN S
PITZEN."
1720 DATA "EINEN GANZ NORMAL
EN WEG      MIT WILDEN PFLANZ
```

```
EN ZU BEI- DEN SEITEN."
1730 DATA "EINE ALTE HAENGEB
RUECKE     UEBER EINEN REISS
ENDEN      FLUSS!"
1740 DATA "EINE WEGBIEGUNG,
DIE ABER   SEHR GEFAEHRLICH
AUSSIEHT!!"
1750 DATA "EINEN HAUFEN ASCH
E. VORHER  STAND HIER SCHEIN
BAR EINMAL EINE HUETTE!!"
1760 DATA "EINE WEGBIEGUNG.
SEITLICH   LIEGT EIN AUSGEHO
EHLTER     BAUMSTAMM."
1770 DATA "EINE WEGGABELUNG.
 NACH      SUEDEN FUEHREN SE
LTSAME     SPUREN..."
1780 DATA "EINE WEGBIEGUNG.
SELTSAME   SPUREN.EVENTUELL
MENSCH-    LICHER NATUR?!?"
1790 DATA "EINE LICHTUNG, DI
E MIT      AUSSERGEWOEHNLICH
 HOHEM GRASBEWACHSEN IST!!"
1800 DATA "DER WEG ENDET HIE
R VOR EINERFELSWAND, DIE MIT
 DICHTEN   PFLANZEN BEWACHSE
N IST."
1810 DATA "EINEN FLUGHAFEN M
IT EINER   UEBERWACHSENEN RU
NWAY UND   VERLASSENEN GEBAE
UDEN!!"
1820 DATA "EINE VERHAELTNISM
AESSIG GUT BEFESTIGTE WEGKRE
UZUNG!"
1830 DATA "EINE WEGBIEGUNG.
DAS GRAS   SEITLICH IST NIED
ERGEDRUECKT"
1840 DATA "DER WEG FUEHRT DU
RCH EINIGE SELTSAME STATUEN.
 UND IMMER DIESE SPUREN..."
1850 DATA "EINE NORMALE WEGB
IEGUNG."
1860 DATA "EINE WEGGABELUNG.
 IM SUEDEN SCHIMMERT ETWAS!!
"
1870 DATA "EINE WEGGABELUNG.
 DER WEG   NACH SUEDEN IST M
IT KLEINEN STEINEN BELEGT."
1880 DATA "EINE WEGGERADE. I
M NORDEN   IST ETWAS..."
1890 DATA "EINE WEGBIEGUNG M
IT SELTSA- MEN PFLANZEN AN D
EN SEITEN??"
1900 DATA "EINE GRUENE WIESE
. IRGEND-  ETWAS PLAETSCHERT
..."
1910 DATA "EINE WEGBIEGUNG.
DER WEG ISTSCHLAMMIG..!"
1920 DATA "EIN SEE MIT EINEM
 FLUSS-    ZULAUF."
1930 DATA "DER FLUSS VERSCHW
INDET      IRENDWO UNTER EIN
ER FELS-   WAND???"
1940 DATA "EINE MIT STEINEN
```

LISTINGS

```
BELEGTE    WEGBIEGUNG."
1950 DATA "DER WEG ENDET HIE
R VOR EINEMGROSSEN STEIN"
1960 REM ** RICHTUNGEN
1970 DATA 0,2,7,0,0,3,0,1,0,
4,0,2,0,0,10,3
1980 DATA 0,6,0,0,0,0,12,5,1
,8,14,0,0,0,15,7
1990 DATA 0,0,16,0,4,0,0,0,0
,12,0,0,6,13,19,11
2000 DATA 0,0,20,12,7,0,21,0
,8,16,0,0
2010 DATA 9,17,0,15,0,18,24,
16,0,19,0,17,12,0,0,18
2020 DATA 13,0,0,0,14,22,0,0
,0,23,0,21,0,0,0,22
2030 DATA 17,25,0,0,0,0,0,24
2040 REM ** GEWONNEN
2050 CALL CLEAR :: PRINT : :
 :"HERZLICHEN GLUECKWUNSCH!!
!!": :"DU HAST DEN SCHATZ GE
HOBEN.": :
2060 PRINT "DAS GOLD IM WERT
 VON": :">> 300.000$ <<": :"
GEHOERT DIR."
2070 PRINT : :"EINSAME SPITZ
E !!!": : :
```

Fortsetzung von Seite 10

Hinweis entlockt werden, der auf ein bestimmtes Dschungelgebiet wies. Eine Versicherungsgesellschaft, die besonders hart von den Diebstählen betroffen wurde, engagiert nun einen Privatdetektiv, nämlich Sie, den Schatz wiederzufinden. Ein Problem ist lediglich die Geldfrage. Die Versicherungsgesellschaft zahlt nämlich nur den Hinflug in das Dschungelgebiet, nicht aber die Ausrüstung.
So finden Sie sich also mutterseelenallein mitten im Dschungel wieder und sollen die Beute finden...
Soweit zur Rahmenhandlung. Zum Programm bleibt zu sagen, daß bei der Eingabe der DATA-Zeilen sehr auf Sorgfalt geachtet werden muß. Die Eingabe der Befehle beim Spiel erfolgt nach dem Schema VERB+NOMEN.
Für das Spiel ist zwar keine Speichererweiterung, wohl aber das Extended-Basic Modul nötig. Die Punkte in diesem Programm diesmal mitabtippen.
So wünsche ich denn viel Spaß auf der Jagd nach dem verlorenen Schatz.

Matthias Renken

LISTING

RETRUN

Zuerst wird das Kurzprogramm mit den Grafikdefinitionen geladen, und gestartet. Es startet dann selbständig das Hauptprogramm von einer Kassette. Das Vorprogramm wird erst wieder benötigt, wenn das Hauptprogramm mit FCTN-CLEAR oder ähnlichem unterbrochen/beendet wurde. Besitzer einer Diskettenstation können den RUN-Befehl im Vorprogramm entsprechend verändern, Besitzer einer Speichererweiterung können beide Programme zu einem zusammenfügen. (Die Trennung war notwendig, da das Hauptprogramm den Speicher vollständig füllt.)

ZUM PROGRAMM SELBST:

Sie müssen mit einem Hubschrauber durch unterirdische Höhlen fliegen. Die Steuerung erfolgt mit den Pfeiltasten. Ein Schuß kann mit ";" ausgelöst werden, wobei dieser unbegrenzt nach unten fällt, wenn der Hubschrauber nach oben/unten fliegt, bzw. stillsteht, oder 5 Schritte zur Seite in Flugrichtung des Hubschraubers. Außerdem steht noch mit "A" eine Bombe zur Verfügung, die in die (erforderliche) Flugrichtung (links/rechts) nach unten fliegt. Sie richtet sich ausschließlich gegen Mauern, wobei sie außer dem getroffenen Stück noch das darüber/darunter bzw. links/rechts liegende Teil mitnimmt (falls vorhanden).
Hat man die Lücke in den Wänden am unteren Ende des Bildes erreicht, so kommt man zum nächsten der drei Bilder, bzw. wieder zum ersten.

DIE GEFAHREN:

1. Der Treibstoffvorrat
Er beträgt am Anfang 2000 Einheiten, die sich laufend vermindern. Man kann ihn jedoch wieder ergänzen, indem man auf dem Deck der Versorgungsstation landet und anschließend X drückt. Wird der Hubschrauber ganz zerstört (siehe unten) so wird der Vorrat nicht(!!!) wieder aufgefüllt.

2. Die Höhlenwände
Mit ihnen sollte man grundsätzlich nicht zusammenstoßen, da dies eine vollständige Zerstörung zur Folge hat. Zu ihnen zählen auch Mauern, unterirdische Seen, usw.

3. Die Vulkane:
Sie brechen in regelmäßigen Zeitintervallen, jedoch in zufälliger Reihenfolge, aus (alle 4 Schritte).
In den Bereich ihrer Auswürfe zu gelangen, hat eine vollständige Vernichtung zur Folge.

4. Vier bewegliche Minen
Sie sind von den unbeweglichen durch ihre grüne Farbe zu unterscheiden. Ein Zusammenstoß mit ihnen hat nur die Vernichtung des Hubschraubers zur Folge, der Pilot springt jedoch ab und kann nun zu Fuß die nächste Versorgungsstation erreichen. Für ihn ist nun jedoch alles tödlich!

DIE STEUERUNG:

S = links
D = rechts
Hierbei werden ein Kästchen hohe Hindernisse automatisch bestiegen.
A = Sprengung des links/rechts/darüber befindlichen Mauerstücks
E = Aktivierung eines Düsenrucksacks (nur hierbei erfolgt Treibstoffverbrauch). Das Männchen steigt nun senkrecht in die Höhe, bis es an der Decke (oder sonstwo) anstößt und fällt anschließend wieder zu Boden. Durch Druck auf S oder D bewegt es sich einen Schritt zur Seite und fällt dann wieder.
Fallen Sie in einen See, so schwimmen Sie automatisch zu einem Boot, das Sie zur Versorgungsstation bringt.
Noch eine Anmerkung zu den Minen: Sie können nicht abgeschossen werden (nur bei beweglichen hat dies Geltung), der Schuß geht durch sie hindurch.

5. Die Lampe
Sie ist die einzige freiwillige Gefahr. Trifft man sie mit einem Schuß, so erhält man zwar 100 Punkte, wird aber unsichtbar. Man kann seinen Aufenthaltsort nur noch anhand des Ursprungs der Schüsse bestimmen. Dies wird erst bei vollständiger oder teilweiser Vernichtung wieder aufgehoben, bzw. bei Erreichen des nächsten Bildes.

Noch ein Hinweis:
Die winkenden Forscher, die Sie laut im Programm enthaltener Anleitung retten sollen, werden durch Berührung eingesammelt.

Zur Eingabe:
Die Großbuchstaben B, C, H und O sind in allen Grafikdefinitionen der folgenden Zeilen zusammen mit der 'CTRL'-Taste einzugeben: 150, 1010, 1020, 1120–1340. Da die Buchstaben auch in Befehlen vorkommen, in denen Text und Grafik gemischt sind, gilt dies nur, wenn die Buchstaben nicht in einem 'sinnvollen' Wort vorkommen.
Es bedeuten:
CTRL B – Luftmine 1
CTRL C – Luftmine 2
CTRL H – Mauer
CTRL O – Lampe

LISTINGS

```
10 ! ************************
11 ! *        RETRUN        *
12 ! *       (Teil 1)       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Heiko  Oehring     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *    Cassettenrec.     *
23 ! *(oder Disk+32K-Erw.)  *
24 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *      931 Bytes       *
28 ! *                      *
29 ! ************************
100 CALL CHAR(96,"FFFFFFF970
302000FFFFDE5C18181000000818
181E7EFFFF181818183C3D7FFF")
110 CALL CHAR(100,"E0FCE0C08
080F0F8F0E0FEF0E0F8F0C00F0F1
FFF7FFF0F07030F3F1F0F7F3F1F"
)
120 CALL CHAR(104,"FFFFFFFFF
FFFFFFF00143E3E7C7EFFFF00000
0000018000000000000000101000"
,108,"0000183C3C180000182419
7B1B3B4810")
130 CALL CHAR(112,"00F8217F9
FF97000007C1038447C3800001F8
4FEF99F0E000080DEFDFFDE8000"
)
140 CALL CHAR(140,"601053BCB
C5310080811 52BCBC521160",143
,"7CFE54547C283810")
150 CALL CHAR(124,"00FFFF404
921100F02070217153F7F557FFFF
F02928408F0",136,"FF89FF91FF
89FF917EFF1824187E1824")
160 CALL CHAR(128,"25124A2A2
AACA9A9AA4A2AAA9C5B3C1C00422
4183C182442187E18FF3C187E18"
)
170 CALL CHAR(116,"80C0E0E0F
8FCFEFF0103070F0F3F7FFF00000
00000000066")
180 CALL CHAR(40,"FFFFFCFEF2
F8CCC2E2C7CEFCF8FCFFFFFFFFBF
1F3F77638743733F9FDF7FFFFF")
190 CALL CHAR(121,"000000000
0FFFEFC1824187E18FFFEFC92543
81038549200",64,"1824183C5A1
82442")
200 RUN "CS1"
```

```
10 ! ************************
11 ! *        RETRUN        *
12 ! *       (Teil 2)       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Heiko  Oehring     *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *    Cassettenrec.     *
23 ! *(oder Disk+32K-Erw.)  *
24 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *      12214 Bytes     *
28 ! *                      *
29 ! ************************
100 CALL CLEAR :: CALL SCREE
N(2):: RANDOMIZE :: CALL COL
OR(1,2,2,2,7,2,9,7,2,10,7,2,
11,5,2,12,13,2,13,11,2,14,7,
16):: DIM VV(11),VH(11)
110 M1$="1824183C5A182442" :
: M2$="1824997E18182442" ::
CALL CHAR(120,M1$):: FOR I=3
 TO 8 :: CALL COLOR(I,11,2):
: NEXT I :: GOTO 160
120 PRINT "iiiiiiiiiiiiiiii
iiiiiiiiiih.................
.........hh hh) hhe fhd hh)
 b c b..c h"
130 PRINT "h h h h....h..h h
 h h h) h hh hhd hd...h..hhd
 h h h*)h h"
140 PRINT "h h h h....h..h h
 *b( h *h hh ` a hhe..`..` a
..a..`..a hh................
..........h"
150 PRINT "h H j p BY H.OEHR
ING r j C hh................
.........h`a`a`a`a`a`a`a`a`
a`a`a`a`a`a": : : :: RETURN
160 B=0 :: GOSUB 120 :: PRIN
T "DRUECKEN SIE:............
...`a`a`a`a`a`a`": :"1 < ANL
EITUNG": :"2 < START": : : :
170 FOR I=30 TO 0 STEP -2 ::
 CALL SOUND(10,110,I,220,I,3
30,I):: CALL SOUND(10,330,I,
440,I,550,I):: CALL SOUND(10
,550,I,660,I,770,I):: NEXT I
180 CALL KEY(0,K,S):: IF K=4
9 THEN 910 ELSE IF K=50 THEN
 190 ELSE 180
190 CALL CLEAR :: CALL COLOR
(1,2,2,11,7,5):: SC,L,B=0 ::
 T=2000 :: CALL SCREEN(7)
200 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL B(VZ,VV(),V
H()):: CALL A(SC):: CALL F(T
):: CALL L(L):: V,H=4
210 CALL SPRITE(#1,113,5,V*8
```

LISTINGS

```
-7,H*8-7):: FOR I=1 TO 4 ::
VM(I)=-I :: NEXT I :: FOR I=
1 TO 4
220 VM(I)=INT(RND*21)+3 :: H
M(I)=INT(RND*30)+2 :: CALL G
CHAR(VM(I),HM(I),Z):: IF Z<>
32 THEN 220
230 FOR O=1 TO 4 :: IF O=I T
HEN 240 ELSE IF VM(O)=VM(I)T
HEN 220
240 NEXT O :: CALL SPRITE(#I
+1,INT(RND*2)+130,13,VM(I)*8
-7,HM(I)*8-7):: RM(I)=-1 ::
NEXT I :: CALL SOUND(500,880
,0,1760,0)
250 FOR I=1 TO 4 :: CALL T(S
F,V,H,SC):: T=T-1 :: CALL F(
T):: IF T=0 THEN 380 ELSE ON
 SF+1 GOTO 260,360,200,380,4
00
260 IF I/2=INT(I/2)THEN CALL
 CHAR(120,M2$)ELSE CALL CHAR
(120,M1$)
270 CALL GCHAR(VM(I),HM(I)+R
M(I),Z):: IF Z=32 THEN 290
280 RM(I)=-RM(I)
290 HM(I)=HM(I)+RM(I):: CALL
 LOCATE(#I+1,VM(I)*8-7,HM(I)
*8-7):: CALL COINC(ALL,X)::
IF X THEN 460
300 NEXT I :: GOSUB 310 :: G
OTO 250
310 X=INT(RND*VZ)+1
320 CALL HCHAR(VV(X)+1,VH(X)
,129):: CALL HCHAR(VV(X),VH(
X),128):: CALL SOUND(-500,-7
,7)
330 IF H<>VH(X)THEN 340 ELSE
 IF V=VV(X)OR V=VV(X)+1 THEN
 350
340 CALL VCHAR(VV(X),VH(X),3
2,2):: RETURN
350 CALL VCHAR(VV(X),VH(X),3
2,2):: GOTO 380
360 CALL SOUND(500,440,0,880
,0):: CALL SOUND(500,880,0,1
760,0):: IF T>20 THEN 370 EL
SE SC=SC+20
370 T=2000 :: CALL A(SC):: C
ALL F(T):: GOTO 300
380 CALL PATTERN(#1,123):: F
OR I=1 TO 30 STEP 5 :: CALL
SOUND(200,-7,I):: NEXT I ::
T=2000 :: L=L+1 :: IF L=5 TH
EN 410 ELSE CALL L(L)
390 V,H=4 :: CALL SPRITE(#1,
113,5,V*8-7,H*8-7):: GOTO 25
0
400 CALL SOUND(-500,440,0)::
 CALL HCHAR(V,H,32):: SC=SC+
50 :: CALL A(SC):: GOTO 250
410 FOR I=0 TO 30 STEP 3 ::
CALL SOUND(250,329,I,165,I):
: CALL SOUND(250,329,I,146,I
):: CALL SOUND(250,262,I,131
,I)
420 CALL SOUND(250,262,I,165
,I):: NEXT I :: CALL DELSPRI
TE(ALL):: CALL CLEAR :: CALL
 SCREEN(2):: CALL COLOR(11,5
,2):: GOSUB 120
430 PRINT "SIE HABEN";SC: :"
PUNKTE ERREICHT": : : :: INP
UT "WOLLEN SIE NOCHMAL?":A$
:: IF SEG$(A$,1,1)="J" THEN
450
440 CALL CLEAR :: END
450 RESTORE :: CALL CLEAR ::
 GOTO 160
460 CALL SPRITE(#1,123,5,V*8
-7,H*8-7):: CALL SOUND(500,-
5,0):: CALL SOUND(1,110,30)
470 CALL PATTERN(#1,137):: G
OSUB 480 :: GOTO 650
480 CALL GCHAR(V+1,H,Z):: IF
 Z<>32 THEN 500
490 CALL SOUND(-100,160,10,1
10,10):: V=V+1 :: CALL LOCAT
E(#1,V*8-7,H*8-7):: GOTO 480
500 IF Z=124 THEN 640 ELSE I
F Z=125 THEN 630 ELSE IF Z=1
21 THEN 580 ELSE IF Z=120 TH
EN 570
510 IF Z=115 OR Z=118 OR Z=1
17 THEN GOSUB 530
520 CALL SOUND(-100,660,0)::
 CALL SOUND(100,880,0):: CAL
L PATTERN(#1,64):: RETURN
530 V=V+1 :: CALL SPRITE(#1,
140,11,V*8-8,H*8-7)
540 CALL SOUND(-500,-5,10)::
 CALL GCHAR(V,H-1,Z):: IF Z=
116 THEN 560 ELSE H=H-1 :: C
ALL SPRITE(#1,141,11,V*8-8,H
*8-7)
550 FOR I=1 TO 10 :: NEXT I
:: CALL PATTERN(#1,140):: FO
R I=1 TO 10 :: NEXT I :: GOT
O 540
560 H=H-1 :: V=V-2 :: GOTO 5
80
570 V=V+1 :: H=H+1 :: CALL S
PRITE(#1,64,5,V*8-7,H*8-7)::
 GOTO 650
580 V=V+1 :: CALL SPRITE(#1,
122,5,V*8-7,H*8-7):: CALL SO
UND(-500,262,0,523,0,1046,0)
590 CALL SOUND(-200,-3,0)::
CALL GCHAR(V,H+1,Z):: IF Z=3
2 THEN 610
600 V=V-1 :: H=H+1 :: CALL S
PRITE(#1,113,5,V*8-7,H*8-7):
: CALL SOUND(-500,440,0,1320
,0,3960,0):: T=2000 :: CALL
F(T):: GOTO 250
610 H=H+1 :: CALL LOCATE(#1,
```

LISTINGS

```
V*8-7,H*8-7):: CALL GCHAR(V+
1,H,Z):: IF Z=117 THEN 620 E
LSE 590
620 H=H+1 :: CALL SPRITE(#1,
64,5,V*8-7,H*8-7):: GOTO 650
630 V=V+1 :: H=H-1 :: CALL L
OCATE(#1,V*8-7,H*8-7)
640 CALL SOUND(500,440,0,132
0,0,3960,0):: CALL PATTERN(#
1,113):: T=2000 :: CALL F(T)
:: GOTO 250
650 FOR I=1 TO 4 :: CALL KEY
(0,K,S):: IF K=65 THEN 710 E
LSE IF K=83 THEN 770 ELSE IF
 K=68 THEN 780 ELSE IF K=69
THEN 850
660 IF I/2=INT(I/2)THEN CALL
 CHAR(120,M2$)ELSE CALL CHAR
(120,M1$)
670 CALL GCHAR(VM(I),HM(I)+R
M(I),Z):: IF Z=32 THEN 690
680 RM(I)=-RM(I)
690 HM(I)=HM(I)+RM(I):: CALL
 LOCATE(#I+1,VM(I)*8-7,HM(I)
*8-7):: CALL COINC(ALL,X)::
IF X THEN 380
700 NEXT I :: GOSUB 310 :: G
OTO 650
710 CALL GCHAR(V+1,H,Z):: IF
 Z<>136 THEN 730 ELSE CALL H
CHAR(V+1,H,123):: CALL SOUND
(-500,-7,0):: SC=SC+15 :: CA
LL A(SC)
720 CALL HCHAR(V+1,H,32):: G
OTO 470
730 X=-1 :: GOTO 750
740 IF X=1 THEN 660 ELSE X=1
750 CALL GCHAR(V,H+X,Z):: IF
 Z<>136 THEN 740 ELSE CALL H
CHAR(V,H+X,123):: CALL SOUND
(-500,-7,0)
760 SC=SC+15 :: CALL A(SC)::
 CALL HCHAR(V,H+X,32):: GOTO
 740
770 X=-1 :: GOTO 790
780 X=1
790 CALL GCHAR(V,H+X,Z):: IF
 Z=32 THEN 810 ELSE IF Z=121
 THEN 840 ELSE IF Z=120 THEN
 830
800 CALL GCHAR(V-1,H+X,Z)::
IF Z<>32 THEN 660 ELSE V=V-1
 :: H=H+X :: CALL LOCATE(#1,
V*8-7,H*8-7):: GOTO 820
810 H=H+X :: CALL LOCATE(#1,
V*8-7,H*8-7):: CALL GCHAR(V+
1,H,Z):: IF Z=32 THEN 470 EL
SE IF Z=117 THEN 530 ELSE IF
 Z=35 THEN 200
820 CALL LOCATE(#1,V*8-7,H*8
-7):: GOTO 660
830 H=H+2*X :: GOTO 820
840 V=V-1 :: H=H+1 :: GOTO 5
80
850 CALL PATTERN(#1,109):: C
ALL SOUND(-4250,-6,10):: T=T
-1 :: CALL F(T):: IF T=0 THE
N 470
860 CALL GCHAR(V-1,H,Z):: IF
 Z<>32 THEN 470 ELSE V=V-1 :
: CALL LOCATE(#1,V*8-7,H*8-7
)
870 CALL KEY(0,K,S):: IF K=8
3 THEN 880 ELSE IF K=68 THEN
 890 ELSE 850
880 X=-1 :: GOTO 900
890 X=1
900 CALL GCHAR(V,H+X,Z):: IF
 Z<>32 THEN 850 ELSE H=H+X :
: CALL LOCATE(#1,V*8-7,H*8-7
):: GOTO 470
910 CALL CLEAR :: CALL COLOR
(1,7,2):: DISPLAY AT(1,1):".
.........RETRUN.........
.........+======)": : :
920 V=3 :: A$="SIE MUESSEN I
M AUFTRAG DER" :: CALL S(V,A
$):: A$="SURYKANT COMPANY IN
 EINEM" :: CALL S(V,A$)
930 A$="UNTERIRDISCHEN HOEHL
ENSYSTEM" :: CALL S(V,A$)::
A$="NACH VERSCHOLLENEN FORSC
HERN" :: CALL S(V,A$)
940 A$="SUCHEN." :: CALL S(V
,A$)
950 V=V+1 :: A$="SIE WERDEN
DABEI VIELEN" :: CALL S(V,A$
)):: A$="GEFAHREN BEGEGNEN,DI
E SIE" :: CALL S(V,A$)
960 A$="UEBERWINDEN MUESSEN.
" :: CALL S(V,A$):: V=V+1 ::
 A$="ACHTEN SIE DABEI JEDOCH
 AUF" :: CALL S(V,A$)
970 A$="IHREN TREIBSTOFFVORR
AT!!" :: CALL S(V,A$):: V=V+
1 :: A$="WIRD IHR HUBSCHRAUB
ER" :: CALL S(V,A$)
980 A$="GETROFFEN,SO MUESSEN
 SIE ZU" :: CALL S(V,A$):: A
$="FUSS DIE NAECHSTE VER-" :
: CALL S(V,A$)
990 A$="SORGUNGSSTATION ERRE
ICHEN." :: CALL S(V,A$):: V=
23 :: A$=" IRGEND EINE TASTE
 DRUECKEN" :: CALL S(V,A$)
1000 CALL KEY(0,K,S):: IF S=
0 THEN 1000 ELSE CALL CLEAR
1010 V=0 :: A$="r = IHR HUBS
CHRAUBER" :: CALL S(V,A$)::
V=V+1 :: A$="H = MAUER" :: C
ALL S(V,A$)
1020 V=V+1 :: A$="O = LAMPE"
 :: CALL S(V,A$):: V=V+1 ::
A$="i = VULKAN" :: CALL S(V,
A$):: V=V+1 :: A$="B C = LUF
TMINEN" :: CALL S(V,A$)
```

LISTINGS

```
1030 V=V+1 :: A$=" }" :: CAL
L S(V,A$):: A$="!~ = VERSORG
UNGSSTATION" :: CALL S(V,A$)
1040 V=V+1 :: A$="y = BOOT"
:: CALL S(V,A$)
1050 V=20 :: A$=".......VIEL
 GLUECK!!!" :: CALL S(V,A$):
: V=23 :: A$=" IRGEND EINE T
ASTE DRUECKEN" :: CALL S(V,A
$)
1060 CALL KEY(0,K,S):: IF S=
0 THEN 1060 ELSE CALL CLEAR
:: GOTO 190
1070 SUB B(VZ,VV(),VH()):: F
OR I=1 TO 3 :: CALL SOUND(15
0,I*110,0):: NEXT I :: CALL
SOUND(150,550,0):: CALL SOUN
D(150,440,0)
1080 CALL SOUND(300,550,0)::
 B=B+1 :: IF B=4 THEN 1360
1090 CALL VCHAR(1,31,31,96):
: DISPLAY AT(1,1):"PUNKTE:..
....FUEL:.........h(`a`a`a`
a`a`a`a`a`a`a`a*h(.........
.................*"
1100 FOR I=4 TO 22 STEP 3 ::
 READ R$ :: DISPLAY AT(I,1):
R$ :: NEXT I
1110 READ VZ :: FOR I=1 TO V
Z :: READ VV(I),VH(I):: NEXT
 I :: SUBEXIT
1120 DATA "d..........B.....
.........fe....C...........
.........gd............C.
.........f"
1130 DATA "e......i....x....
i.........g)bcbcbchcbcbHbcbc
hcbcb)....+hhhhhhhhhhhhhhhhh
hhhhhhHHHHh"
1140 DATA "(a`a`a`a`a`a`a`a
a`a`a(....*d...............
.........fe.....B..........
.C........g"
1150 DATA "d........i x....i
....i.....fe....bccbhcHcbcbh
cbbchcbbcb+d....a``aa`a`a`a`
`a`aa`a`a(*"
1160 DATA "e.......B........
.........gd...............
C..C......fe..............
.}........g"
1170 DATA "d...i..y........
!~........f)bcchb+tvvvvsvvvv
vvvvvu)...g(a`a`aa`a`a``a`a`
`a`a`a`...f"
1180 DATA "e.......B........
.........gd............C...
.x........fe###bcbbccicbcbcc
bHbbicbbcc+"
1190 DATA 8,5,10,5,20,11,12,
11,19,11,24,17,7,22,13,22,24
1200 DATA ").........C...bcb
...C cbc..gh) B...B......*h(
 B...`aa..fhh).........B...H.
.........g"
1210 DATA "hh(..i.........O.
.C..i..B..fh(..+h)....x..+h)
...+h)..x +(..+hhh)cicHb+(`*
)c+(a*)bH+h"
1220 DATA "d..`a`aa`a``*(...
`*(...*(a`*)..x.....B.....}C
.........fh)+H)y........!~.
...iB...B g"
1230 DATA "hhhhhtvvvvvsvvvvv
svuh)bi...+hhh(`a`a`a`*hhhhh
hhhhhhhHHHhhh(.........*hhh(
`a`a`*(...*"
1240 DATA "h(..C........a`a.
..B.......f(.....B..i B.....
.........ge.......+h)...C..
bcbc x..C f"
1250 DATA "d C +)b+hhh)....+
hhhh)Hcicb+e...*hhhhhhhHHHHh
hhhhhh(`a`*d....aa``a`a....a
`a`aa`....+"
1260 DATA ") B..........B...
.......C +hh)...C..........C
..x.....+hhhh)cbccbbibc####c
bcbccbi+hhh"
1270 DATA 9,5,8,5,24,7,12,10
,23,11,26,15,12,17,27,22,12,
22,26
1280 DATA ")..bbcbcc...C x..
..B.......gh)+hhhhhh)bcbcHci
bccbcbbi..*h`a`a``a`aa`a``aa
`a`a*hhhHHO"
1290 DATA "(..........B....C
.....*h(..*e....b xCbcb...b.
...C..H...fdC..+h)H+(a*)i+(
bibc..H C g"
1300 DATA "e B *h(a`...a`a..
*hhh) H...fd....`..........
.a*hh)h)..f).........i......i
...*hhhhHHh"
1310 DATA "h)....x +h)....+h
)...*hh(..*hh)bicH+hhh)..+hh
h)B..*(...+ha`a``a`a*hhHHhhh
hh)b.....+h"
1320 DATA "(.........*(..*h(
*hhh)cbi+hhd B..C.........`
gh(*hh(a`*hd......}..B......
+(..*(B...*"
1330 DATA "e...y !~.....B..+
(...fh) C gdC ftvvvvvsvvvvu(
.....*h)..+)...`a`a*h(a`a`a.
......*( fh"
1340 DATA "h)....x..H B....C
..i.......*hh)i CH..H.......
.+h)..C...ghhhhb+h)+h)bbibcb
+hhh)######f"
1350 DATA 11,3,19,3,27,7,16,
7,21,10,12,10,19,12,7,14,27,
20,22,21,6,22,16
1360 B=1 :: RESTORE 1120 ::
GOTO 1090 :: SUBEND
1370 SUB T(SF,V,H,SC):: CALL
```

```
 KEY(0,K,S):: IF S<>0 THEN 1
380 ELSE 1400
1380 IF K=59 THEN 1540 :: IF
 K=65 THEN 1690
1390 IF K=69 THEN 1410 :: IF
 K=83 THEN 1430 :: IF K=88 T
HEN 1460 :: IF K=68 THEN 150
0
1400 CALL PATTERN(#1,113)::
FR,SF=0 :: SUBEXIT
1410 CALL GCHAR(V-1,H,Z):: I
F Z<>32 THEN SF=3 ELSE SF=0
1420 FR=0 :: CALL PATTERN(#1
,113):: V=V-1 :: GOTO 1530
1430 CALL GCHAR(V,H-1,Z):: I
F Z<>32 THEN SF=3
1440 IF Z=120 THEN SF=4 ELSE
 IF Z=32 THEN SF=0
1450 FR=-1 :: H=H-1 :: CALL
PATTERN(#1,112):: GOTO 1530
1460 IF V=24 THEN 1530 ELSE
CALL GCHAR(V+1,H,Z):: IF Z<>
32 THEN SF=3 :: IF Z=120 THE
N SF=4
1470 IF Z=35 THEN SF=2 ELSE
IF Z=32 THEN SF=0 ELSE IF Z=
120 THEN SF=4
1480 IF Z=124 THEN SF=1 :: S
UBEXIT
1490 FR=0 :: V=V+1 :: CALL P
ATTERN(#1,113):: GOTO 1530
1500 CALL GCHAR(V,H+1,Z):: I
F Z<>32 THEN SF=3
1510 IF Z=120 THEN SF=4 ELSE
 IF Z=32 THEN SF=0
1520 FR=1 :: H=H+1 :: CALL P
ATTERN(#1,114)
1530 CALL LOCATE(#1,V*8-7,H*
8-7):: SUBEXIT
1540 SF=0 :: IF FR=0 THEN 16
20
1550 FOR I=1 TO 5 :: CALL GC
HAR(V,H+I*SGN(FR),Z):: CALL
HCHAR(V,H+I*SGN(FR),106)
1560 IF Z=130 OR Z=131 THEN
1590 ELSE IF Z=136 THEN 1610
 ELSE IF Z=143 THEN 1850 ELS
E IF Z<>32 THEN 1580
1570 CALL HCHAR(V,H+I*SGN(FR
),Z):: NEXT I :: SUBEXIT
1580 CALL SOUND(-10,-1,0)::
CALL HCHAR(V,H+I*SGN(FR),Z):
: SUBEXIT
1590 CALL HCHAR(V,H+I*SGN(FR
),123):: CALL SOUND(-500,-5,
0):: SC=SC+25
1600 CALL A(SC):: CALL HCHAR
(V,H+I*SGN(FR),32):: SUBEXIT
1610 CALL HCHAR(V,H+I*SGN(FR
),123):: CALL SOUND(-500,-5,
0):: SC=SC+15 :: GOTO 1600
1620 IF V+FR=24 THEN 1680 EL
SE FR=FR+1 :: CALL GCHAR(V+F
R,H,Z):: CALL HCHAR(V+FR,H,1
07)
1630 IF Z=130 OR Z=131 THEN
1650 ELSE IF Z=136 THEN 1670
 ELSE IF Z=143 THEN 1860 ELS
E IF Z<>32 THEN 1680
1640 CALL HCHAR(V+FR,H,Z)::
GOTO 1620
1650 CALL HCHAR(V+FR,H,123):
: CALL SOUND(-500,-5,0):: SC
=SC+25
1660 CALL A(SC):: CALL HCHAR
(V+FR,H,32):: FR=0 :: SUBEXI
T
1670 CALL HCHAR(V+FR,H,123):
: CALL SOUND(-500,-5,0):: SC
=SC+15 :: GOTO 1660
1680 CALL SOUND(-10,-1,0)::
CALL HCHAR(V+FR,H,Z):: FR=0
:: SUBEXIT
1690 Y=FR :: IF FR=0 THEN SU
BEXIT ELSE FR,X,SF=0
1700 IF X=24 THEN 1730 ELSE
X=X+1 :: CALL GCHAR(V+X,H+FR
,Z):: CALL HCHAR(V+X,H+FR,10
8)
1710 IF Z=136 THEN 1740 ELSE
 IF Z=32 THEN 1720 ELSE 1730
1720 CALL HCHAR(V+X,H+FR,32)
:: FR=FR+SGN(Y):: GOTO 1700
1730 CALL HCHAR(V+X,H+FR,Z):
: CALL SOUND(-10,-1,0):: SUB
EXIT
1740 CALL HCHAR(V+X,H+FR,123
):: CALL SOUND(500,-5,0):: S
C=SC+15
1750 CALL HCHAR(V+X,H+FR,32)
:: CALL GCHAR(V+X-1,H+FR,Z):
: IF Z<>136 THEN 1770 ELSE C
ALL HCHAR(V+X-1,H+FR,123)
1760 SC=SC+15 :: CALL HCHAR(
V+X-1,H+FR,32)
1770 CALL GCHAR(V+X,H+FR-1,Z
):: IF Z<>136 THEN 1790 ELSE
 CALL HCHAR(V+X,H+FR-1,123)
1780 SC=SC+15 :: CALL HCHAR(
V+X,H+FR-1,32)
1790 CALL GCHAR(V+X,H+FR+1,Z
):: IF Z<>136 THEN 1810 ELSE
 CALL HCHAR(V+X,H+FR+1,123)
1800 SC=SC+15 :: CALL HCHAR(
V+X,H+FR+1,32)
1810 IF V+X=24 THEN 1830 ELS
E CALL GCHAR(V+X+1,H+FR,Z)
1820 IF Z<>136 THEN 1830 ELS
E CALL HCHAR(V+X+1,H+FR,123)
:: SC=SC+15 :: CALL HCHAR(V+
X+1,H+FR,32)
1830 CALL A(SC):: Y=0 :: SUB
EXIT
1840 CALL SOUND(100,262,0,44
0,0,523,0):: SC=SC+100 :: CA
LL A(SC):: CALL COLOR(#1,2):
```

LISTING

```
: SUBEXIT
1850 CALL HCHAR(V,H+I*SGN(FR
),32):: GOTO 1840
1860 CALL HCHAR(V+FR,H,32)::
 GOTO 1840 :: SUBEND
1870 SUB S(V,A$):: V=V+1 ::
FOR I=1 TO LEN(A$):: CALL HC
HAR(V,I+2,114):: FOR O=1 TO
10 :: NEXT O
1880 DISPLAY AT(V,I):SEG$(A$
,I,1):: NEXT I :: SUBEND
1890 SUB A(SC):: DISPLAY AT(
1,9)SIZE(4):USING "####":SC
:: SUBEND
1900 SUB L(L):: CALL HCHAR(1
,27,32,4):: CALL HCHAR(1,27,
64,4-L):: SUBEND
1910 SUB F(T):: DISPLAY AT(1
,20)SIZE(4):USING "####":T :
: SUBEND
```

NETZTEIL-BERECHNUNG

Netzteilberechnung ist ein Anwenderprogramm für Hobby-Elektroniker.
Das Programm ist menügesteuert.
Nach Eingabe von Ampere und Volt zeichnet der TI einen Schaltplan von einem Netzteil.
Anschließend werden die Werte von Trafo, Gleichrichter, Kondensatoren und IC-Spannungsregler ausgegeben.

RESONANZ-FREQUENZ

Mit diesem menügesteuerten Programm ist unseren Lötkolbenfreaks und all denjenigen, die mit Schwingungskreisen arbeiten, die Möglichkeit gegeben, ihre Berechnungen direkt am Bildschirm auszuführen.
Versuche und Änderungen, bis das optimale Ergebnis erreicht ist, gehen nun viel schneller von der Hand, als wenn sie "zu Fuß" durchgeführt werden müßten.
Je nachdem, welche Daten bekannt sind, können die restlichen Werte berechnet werden.
Mit Hilfe eines Unterprogrammes können Spulen entworfen und deren theoretische Werte abgelesen werden. Daneben ist ein weiteres Unterprogramm eingebaut, mit dem sich der Ersatzwiderstand von zwei parallelgeschalteten Widerständen berechnen läßt. Hierbei ist es auch möglich, die Buchstaben "K" und "M" zu benutzen, zum Beispiel 12K oder 2.2M.
Das Programm ist vollständig in TI-Basic geschrieben und erfordert deshalb außer einem Kassettenrecorder zur Programmabspeicherung keinerlei Zubehör.

LISTINGS

```
100 !************************
110 !*                      *
120 !*  NETZTEILBERECHNUNG  *
130 !*                      *
140 !*    COPYRIGHT BY      *
150 !*                      *
160 !*     REINER WOLF      *
170 !*                      *
180 !*                      *
190 !*                      *
200 !*  BENOETIGTE GERAETE  *
210 !*   TI99/4A KONSOLE    *
220 !*     EXT. BASIC       *
230 !*                      *
240 !*                      *
250 !*  SPEICHERBELEGUNG    *
260 !*      4390 BYTS       *
270 !*                      *
280 !************************
290 ON ERROR 1150
300 CALL CHAR(117,"0000444444447A40",96,"010A043A18284080",97,"8050205C18140201",98,"8040281B3A040A01",99,"0102141B5C205080",100,"9999599A599A599A")
310 CALL CHAR(101,"599A599A599A9999",104,"FF81FF0000FFFFFF",108,"18244295A9422418",126,"000000FF",127,"8080808080808080")
320 CALL CHAR(128,"808080FF80808080",129,"000000FF80808080",130,"808080FF",131,"0000000808080808080",132,"80808080")
330 CALL MAGNIFY(2)
340 CALL CLEAR
350 CALL SCREEN(5):: FOR I=0 TO 14 :: CALL COLOR(I,16,5) :: NEXT I
360 DISPLAY AT(1,8):"NETZTEIL MIT" :: DISPLAY AT(3,5):"IC SPANNUNGSREGLER"
370 DISPLAY AT(6,1):"BITTE STROM EINGEBEN:   AMP."
380 DISPLAY AT(8,3):"MOEGLICH SIND 0.1,0.5," :: DISPLAY AT(10,17):"1,2," :: DISPLAY AT(12,17):"5,10 AMP."
390 ACCEPT AT(6,22)BEEP SIZE(3)VALIDATE(NUMERIC,"."):I
400 IF I=0.1 OR I=.5 OR I=1 OR I=2 OR I=5 OR I=10 THEN 410 ELSE 390
410 DISPLAY AT(15,1):"BITTE SPANNUNG EINGEBEN:   V"
420 DISPLAY AT(17,3):"MOEGLICH SIND 5,6,8," :: DISPLAY AT(19,17):"12,15," :: DISPLAY AT(21,17):"18,24 VOLT"
430 ACCEPT AT(15,25)BEEP SIZE(2)VALIDATE(DIGIT):U
440 IF U=5 OR U=6 OR U=8 OR U=12 OR U=15 OR U=18 OR U=24 THEN 450 ELSE 430
450 CALL CLEAR :: GOTO 870
460 IF I=.1 THEN 530
470 IF I=.5 THEN 610
480 IF I=1 THEN 690
490 IF I=2 THEN 770
500 IF I=5 THEN 820
510 IF I=10 THEN 850
520 GOTO 340
530 IF U=5 THEN A$="78L05"
540 IF U=6 THEN A$="78L06"
550 IF U=8 THEN A$="78L08"
560 IF U=12 THEN A$="78L12"
570 IF U=15 THEN A$="78L15"
580 IF U=18 THEN A$="78L18"
590 IF U=24 THEN A$="78L24"
600 RETURN
610 IF U=5 THEN A$="78M05"
620 IF U=6 THEN A$="78M06"
630 IF U=8 THEN A$="78M08"
640 IF U=12 THEN A$="78M12"
650 IF U=15 THEN A$="78M15"
660 IF U=18 THEN A$="78M18"
670 IF U=24 THEN A$="78M24"
680 RETURN
690 IF U=5 THEN A$="7805"
700 IF U=6 THEN A$="7806"
710 IF U=8 THEN A$="7808"
720 IF U=12 THEN A$="7812"
730 IF U=15 THEN A$="7815"
740 IF U=18 THEN A$="7818"
750 IF U=24 THEN A$="7824"
760 RETURN
770 IF U=5 THEN A$="78S05"
780 IF U=12 THEN A$="78S12"
790 IF U=15 THEN A$="78S15"
800 IF U=24 THEN A$="78S24"
810 RETURN
820 IF U=5 THEN A$="78H12KC"
830 IF U=12 THEN A$="78H12KC"
840 RETURN
850 IF U=5 THEN A$="78P05KC"
860 RETURN
870 CALL HCHAR(6,3,108):: CALL HCHAR(10,3,108):: CALL HCHAR(6,4,126):: CALL HCHAR(10,4,126)
880 CALL SPRITE(#1,100,16,45,34,#2,101,16,61,34)
890 CALL HCHAR(6,7,126,3):: CALL HCHAR(10,7,126,3)
900 CALL SPRITE(#3,96,16,45,57,#4,97,16,61,57)
910 DISPLAY AT(5,3):"TR  GL"
920 CALL SPRITE(#5,98,16,45,73,#6,99,16,61,73)
930 CALL HCHAR(8,12,126,10):: CALL VCHAR(8,8,127,2):: CALL HCHAR(10,8,128):: CALL VCHAR(11,8,127,7)
```

```
940 CALL SPRITE(#7,104,16,89
,98,#8,104,16,89,138):: DISP
LAY AT(11,13):"+    +"
950 DISPLAY AT(13,9):"C1   C
2"
960 CALL HCHAR(8,14,129):: C
ALL VCHAR(9,14,127,3):: CALL
 HCHAR(8,19,129):: CALL VCHA
R(9,19,127,3):: CALL HCHAR(1
7,8,130)
970 CALL HCHAR(17,9,126,21):
: CALL VCHAR(14,14,127,3)::
CALL HCHAR(17,14,130):: CALL
 VCHAR(14,19,127,3):: CALL H
CHAR(17,19,130)
980 CALL HCHAR(6,22,126,5)::
 CALL HCHAR(10,22,126,5):: C
ALL HCHAR(6,22,129):: CALL V
CHAR(7,22,127,3):: CALL HCHA
R(10,22,130)
990 CALL HCHAR(6,27,131):: C
ALL VCHAR(7,27,127,3):: CALL
 HCHAR(10,27,132):: DISPLAY
AT(7,19)SIZE(1):"E" :: DISPL
AY AT(7,26):"A"
1000 CALL HCHAR(11,23,45)
1010 DISPLAY AT(7,21)SIZE(3)
:"IC1" :: DISPLAY AT(9,21)SI
ZE(4):"78XX"
1020 CALL HCHAR(10,24,129)::
 CALL VCHAR(11,24,127,6):: C
ALL HCHAR(17,24,130):: CALL
HCHAR(8,27,128)
1030 CALL HCHAR(8,28,126,2):
: CALL HCHAR(8,31,43):: CALL
 HCHAR(17,31,45)
1040 DISPLAY AT(1,1):"TRAFO"
;U+3;"VOLT,"
1050 DISPLAY AT(1,15):I;"AMP
ERE"
1060 DISPLAY AT(3,1):"GLEICH
RICHTER 40 V";I;"AMP."
1070 DISPLAY AT(18,1):"C1-EL
KO MIN.";U;"V"
1080 DISPLAY AT(18,18):I*200
0;"uF"
1090 DISPLAY AT(20,1):"C2-EL
KO MIN.";U;"V  10 uF"
1100 GOSUB 460
1110 DISPLAY AT(22,1):"IC1:
";A$
1120 DISPLAY AT(22,13):U;"V
";I;"AMP."
1130 DISPLAY AT(24,4):"NEUE
BERECHNUNG J/N"
1140 CALL KEY(0,K,S):: IF S=
0 THEN 1140 :: IF K=74 THEN
1150 ELSE IF K=78 THEN 1160
1150 CALL DELSPRITE(ALL):: A
$="" :: GOTO 340
1160 CALL CLEAR :: RUN "" ::
 END
```

LISTINGS

```
10 REM*********************
11 REM* RESONANZFREQUENZ *
12 REM*                  *
13 REM*  Copyright  by   *
14 REM*                  *
15 REM*   JAN KUENEN     *
16 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                  *
23 REM* Speicherbelegung *
24 REM*    3749 Bytes    *
25 REM*********************
100 CALL SCREEN(15)
110 CALL CLEAR
120 CALL CHAR(118,"0000002424
242458")
130 CALL CHAR(117,"0028004444
4444438")
140 CALL CHAR(97,"0028003844
7C4444")
150 CALL CHAR(120,"FF0000FFF
F0000FF")
160 CALL CHAR(128,"0038448282
24444C6")
170 CALL COLOR(12,5,8)
180 K$(0)="xxxxxxxxxxxxxxxx
xxxxxxxxxx"
190 K$(1)="1.RESONANZFREQUEN
Z BERECHNEN"
200 K$(2)="2.UNBEKANNTE SPUL
E BERECHNEN"
210 K$(3)="3.KONDENSATOR BER
ECHNEN"
220 K$(4)="4.KAPAZITIVE REAK
TANZ"
230 K$(5)="5.SPULEN MACHEN"
240 K$(6)="6.INDUKTIVE REAKT
ANZ"
250 K$(7)="7.ERSATZWIDERSTAN
D BERECHNEN"
260 K$(8)="8.DC-WIDERSTAND D
ER SPULE"
270 FR$="FREQUENZ......MHZ:
"
280 SP$="SPULE.........vH:
"
290 KO$="KONDENSATOR....pF:
"
300 DS$="DIAMETER SPULE MM:
"
310 WI$="WINDUNGEN........:
"
320 DR$="DRAHTDIAMETER..MM:
"
330 RE$="REAKTANZ.........:"
340 CALL CLEAR
350 PRINT K$(0): :TAB(12);"M
ENu": :
360 FOR I=0 TO 8
370 PRINT K$(I): :
380 NEXT I
390 PRINT K$(0)
```

LISTINGS

```
400 CALL KEY(0,K,S)
410 A=K-48
420 IF (A>8)+(A<1)THEN 400
430 CALL CLEAR
440 L=INT((30-LEN(K$(A)))/2)
450 PRINT K$(0): :TAB(L);K$(
A): :K$(0): :
460 ON A GOSUB 530,620,710,8
00,890,1120,1210,1570
470 GOTO 340
480 PRINT K$(0): :"M=MENu...
.....W=WIEDERHOLEN"
490 CALL KEY(0,K,S)
500 IF (K<>87)*(K<>77)THEN 4
90
510 CALL HCHAR(23,1,32,32)
520 RETURN
530 INPUT KO$:C
540 PRINT
550 INPUT SP$:L
560 PRINT
570 F=INT((1000/((8*ATN(1))*
(SQR(L*C))))*1000)/1000
580 PRINT FR$;STR$(F): :
590 GOSUB 480
600 IF K=87 THEN 530
610 RETURN
620 INPUT FR$:F
630 PRINT
640 INPUT KO$:C
650 PRINT
660 L=INT(25330296/(C*(F^2))
)/1000
670 PRINT SP$;STR$(L): :
680 GOSUB 480
690 IF K=87 THEN 620
700 RETURN
710 INPUT FR$:F
720 PRINT
730 INPUT SP$:L
740 PRINT
750 C=INT(253302.96/(L*(F^2)
))/10
760 PRINT KO$;STR$(C): :
770 GOSUB 480
780 IF K=87 THEN 710
790 RETURN
800 INPUT FR$:F
810 PRINT
820 INPUT KO$:C
830 PRINT
840 XC=INT(10^7/(8*ATN(1)*F*
C))/10
850 PRINT RE$;XC;CHR$(128):
:
860 GOSUB 480
870 IF K=87 THEN 800
880 RETURN
890 INPUT DS$:D
900 PRINT
910 INPUT WI$:W
920 PRINT
930 INPUT DR$:DD
940 PRINT
950 LS=DD*W
960 R1=D/2
970 L=(1/25)*(((R1^2)*(W^2))
/(9*R1+10*LS))
980 L=INT(L*1000)/1000
990 PRINT "INDUKTIVITaT...vH
:";L: :
1000 PRINT "SPULENLaNGE....M
M:";LS: :
1010 INPUT KO$:C
1020 PRINT
1030 F=1/((8*ATN(1))*SQR(L*C
))
1040 F=INT(F*10^5)/100
1050 PRINT FR$;STR$(F): :
1060 XL=8*ATN(1)*F*L
1070 XL=INT(XL+.5)
1080 PRINT RE$;XL;CHR$(128):
:
1090 GOSUB 480
1100 IF K=87 THEN 890
1110 RETURN
1120 INPUT FR$:F
1130 PRINT
1140 INPUT SP$:L
1150 PRINT
1160 XL=INT((8*ATN(1)*F*L)/1
00+.5)/10
1170 PRINT RE$;XL;CHR$(128):
:
1180 GOSUB 480
1190 IF K=87 THEN 1120
1200 RETURN
1210 FOR J=1 TO 2
1220 CALL HCHAR(23,1,32,32)
1230 PRINT "WIDERSTAND";J;
1240 INPUT "...:.":A$
1250 D=0
1260 Q=0
1270 FOR I=1 TO LEN(A$)
1280 B$=SEG$(A$,I,1)
1290 IF B$<>"." THEN 1330
1300 IF Q=1 THEN 1220
1310 Q=1
1320 GOTO 1400
1330 IF B$<CHR$(48)THEN 1220
1340 IF B$<CHR$(58)THEN 1400
1350 IF B$="K" THEN 1380
1360 IF B$<>"M" THEN 1220
1370 D=3
1380 D=D+3
1390 A$=SEG$(A$,1,LEN(A$)-1)
1400 NEXT I
1410 X(J)=VAL(A$)*10^D
1420 PRINT
1430 NEXT J
1440 R=(X(1)*X(2))/(X(1)+X(2
))
1450 T=128
1460 IF R<1000 THEN 1520
1470 T=75
1480 IF R<10^6 THEN 1510
```

```
1490 T=T+2
1500 R=R/1000
1510 R=R/1000
1520 R=INT(R*10+.5)/10
1530 PRINT "ERSATZWIDERSTAND
:";R;CHR$(T): :
1540 GOSUB 480
1550 IF K=87 THEN 1210
1560 RETURN
1570 INPUT DS$:DM
1580 PRINT
1590 INPUT DR$:D
1600 PRINT
1610 INPUT "SILBER/KUPFER S-
K: ":A$
1620 PRINT
1630 Z=61.35
1640 IF SEG$(A$,1,1)="S" THE
N 1670
1650 IF SEG$(A$,1,1)<>"K" TH
EN 1610
1660 Z=56
1670 INPUT WI$:W
1680 PRINT
1690 L=(((8*ATN(1)*DM)+2*D)*
W+20)/1000
1700 ML=INT(L*100+.5)
1710 PRINT "DRAHTLaNGE.....
.:";ML;"ZM": :
1720 R=L/(D*Z)
1730 PRINT "WIDERSTAND SPULE
 :";INT(R*1000+.5)/1000;CHR$
(128): :
1740 GOSUB 480
1750 IF K=87 THEN 1570
1760 RETURN
```

LISTINGS

VIER-GEWINNT

Bereits vor drei Jahren wurde von Herrn Alexander Berger ein Spielprogramm gleichen Namens in einem Sonderband einer Computerzeitschrift veröffentlicht.
Da wir nicht annehmen, daß jeder TI-User diesen Sonderband besitzt, entschlossen wir uns, eine ähnliche Version dieses Strategie-Spiels zu veröffentlichen, um eine eventuelle Lücke in Euerer Programmbibliothek zu schließen.
Dieses Programm hat gegenüber der Erstveröffentlichung sogar noch den Vorteil, daß wahlweise gegen einen zweiten Mitspieler oder gegen den Computer angetreten werden kann.
Beim Abtippen ist besondere Sorgfalt auf die DATA-Zeilen zu legen, da das Spiel bei Falscheingaben unter Umständen zwar laufen würde, der Rechner aber einige 4er-Reihen nicht richtig erkennen könnte.
Die Spielregeln, das Spielbrett und die Spielscheiben sind möglichst originalgetreu programmiert worden.
Die Spieler setzen abwechselnd ihre roten bzw. gelben Scheiben in eine der sieben Spalten, von wo aus die Scheibe nach unten auf den Boden oder auf eine vorherige Scheibe fällt.
Gewonnen hat derjenige, der vier Scheiben seiner Farbe in eine Reihe waagerecht, senkrecht oder diagonal zusammenbekommen hat.
Sind alle Spalten voll besetzt und keiner der Spieler hat sein Ziel erreicht, dann endet die Partie unentschieden.
Für die Zugeingabe muß nur die Zifferntaste der gewünschten Spalte gedrückt werden. Der Computer führt den Zug aus, wenn die Eingabe richtig ist und überprüft, ob sich vier Steine einer Farbe in einer Reihe befinden.
Will man beim Spiel gegen den Computer die Farben tauschen, so drückt man einfach die Enter-Taste, wenn der Computer nach einem Zug fragt. Auf diese Weise kann man auch den Computer anfangen lassen, indem man bei der Frage nach dem ersten Zug 'Enter' drückt.
Die Bedenkzeit des Computer beträgt pro Zug 30 s.

LISTINGS

```
100 REM **********************
      *                      *
      *                      *
      *   4 GEWINNT V2.5     *
      *                      *
110 REM *                    *
      *   COPYRIGHT BY       *
      *                      *
      *                      *
      *(c)JENS TIEDEMANN     *
120 REM *                    *
      *                      *
      *BENOETIGTE GERAETE*
      * TI99/4A  KONSOLE    *
      *    EXT. BASIC        *
130 REM *                    *
      *                      *
      * SPEICHERBELEGUNG     *
      *                      *
      *   10302 BYTES        *
140 REM *                    *
      *                      *
      **********************
150 CALL CLEAR :: CALL MAGNI
FY(3):: RANDOMIZE
160 DIM F$(6,7),M$(69),VI(69
,2)
170 CHA=136 :: ZUG=0 :: COL$
(1)="GELB" :: COL$(2)=" ROT"
180 CALL TITEL
190 DATA 093951,10405157,114
1515763,1221425157636 9,22435
76369,23446369,244569
200 DATA 05323950,0609334050
56,0710213441505662,08111722
35425056626 8,121823364356626
8,192437446268,20384568
210 DATA 0125323949,02052126
33404955,0306091722273441495
561,0407101318232835424955 61
67,0811141924293643556167,12
152030374461 67,1631384567
220 DATA 2125323948,01172226
33404854,0205131823273441485
460,0306091419242835424854 60
66,0407101520293643546066,08
111630374460 66,1231384566
230 DATA 17253247,1318263347
53,0114192734475359,02051520
283547535965,0306162936535 96
5,040730375965,08313865
240 DATA 132546,14264652,152
7465258,0116284652586 4,02295
25864,03305864,043164
250 DATA 311,321,331,341,211
,221,231,241,111,121,131,141
,342,352,362,372,242,252,262
,272,142,152,162
260 DATA 172,313,323,333,343
,353,363,373,213,223,233,243
,253,263,273,113,123,133,143
,153,163,173,614
270 DATA 514,414,314,214,114
,624,524,424,324,224,124,634
,534,434,334,234,134,644,544
,444,344,244,144
280 RESTORE :: FOR Y=1 TO 6
:: FOR X=1 TO 7 :: READ F$(Y
,X):: NEXT X :: NEXT Y
290 FOR I=1 TO 69 :: READ M$
(I):: NEXT I
300 A$="0003 0F1F3F3F7F7F7F7F
3F3F1F0F03 0000C0F0F8FCFCFEFE
FEFEFCFCF8F0C000"
310 CALL SCREEN(2):: CALL DE
LSPRITE(ALL):: CALL CLEAR ::
 CALL COLOR(13,12,5,14,10,5)
320 CALL COLOR(0,16,1,2,16,1
,3,16,1,4,16,1,5,16,1,6,16,1
,7,16,1,8,16,1,9,1,5,10,1,5)
330 CALL CHAR(96,A$,128,A$,1
36,A$)
340 CALL CHAR(100,"FEFCF8F0E
0E0E0E0E0E0E0E0E0E0E0E0E0000000
01F3F000000",103,"000000FEFC
0000000102040810101010101010
1010101010101010101113171F")
350 DISPLAY AT(3,7):"d";RPT$
("fg",7);"h"
360 FOR I=1 TO 6 :: DISPLAY
AT(2+2*I,7):"e";RPT$("'b",7)
;"i":TAB(7);"e";RPT$("ac",7)
;"i" :: NEXT I :: DISPLAY AT
(17,8):" 1 2 3 4 5 6 7"
370 CALL HCHAR(15,24,106)
380 DISPLAY AT(20,1)BEEP:"1.
 SPIELER GEGEN COMPUTER": :"
2. ZWEI SPIELER": :"IHRE WAH
L ?"
390 CALL KEY(0,K,S):: IF K<4
9 OR K>50 THEN 390
400 CALL HCHAR(20,1,32,160):
: IF K=50 THEN 860
410 IF ZUG>41 THEN DISPLAY A
T(21,9):"UNENTSCHIEDEN" :: R
UN ELSE CHA=264-CHA :: DISPL
AY AT(21,10)BEEP:"IHR ZUG ?"
420 CALL KEY(0,K,S):: IF K=1
3 THEN DISPLAY AT(21,8):"  C
OMPUTING" :: GOTO 530 ELSE I
F K<49 OR K>55 THEN 420
430 CALL HCHAR(21,21,K):: X=
K-48
440 IF Z(X)=6 THEN CALL SOUN
D(100,220,0):: GOTO 420
450 Y,Z(X)=Z(X)+1 :: CALL SE
TZEN(Y,X,CHA):: ZUG=ZUG+1
460 DISPLAY AT(21,8):"  COMP
UTING"
470 Q$=F$(Y,X)
480 FOR I=1 TO LEN(Q$)STEP 2
 :: N=VAL(SEG$(Q$,I,2))
490 VI(N,(CHA-120)/8)=VI(N,(
CHA-120)/8)+1
500 IF VI(N,(CHA-120)/8)=4 T
HEN DISPLAY AT(21,8):COL$((C
```

```
HA-120)/8);" HAT VIER" :: CA
LL VIER(M$(N),CHA):: RUN
510 NEXT I
520 CHA=264-CHA
530 ZUG=ZUG+1 :: FOR X=1 TO
7 :: IF Z(X)=6 THEN 570
540 Q$=F$(Z(X)+1,X)
550 FOR I=1 TO LEN(Q$)STEP 2
 :: N=VAL(SEG$(Q$,I,2)):: IF
 VI(N,(CHA-120)/8)=3 THEN 80
0
560 NEXT I
570 NEXT X
580 SUM=0 :: FOR X=1 TO 7 ::
 IF Z(X)=6 THEN 780
590 V(1),V(2),W(1),W(2),SU(X
)=0 :: Q$=F$(Z(X)+1,X)
600 T$="0" :: FOR I=1 TO LEN
(Q$)STEP 2 :: N=VAL(SEG$(Q$,
I,2))
610 IF VI(N,(144-CHA)/8)=3 T
HEN 800
620 IF VI(N,1)*VI(N,2)>0 THE
N SU(X)=SU(X)+1 :: GOTO 650
630 IF POS(T$,SEG$(M$(N),3,1
),1)<>0 AND VI(N,1)+VI(N,2)<
2 THEN SU(X)=SU(X)+1 :: GOTO
 650
640 SU(X)=SU(X)+(VI(N,1)+1)^
2+(VI(N,2)+1)^2 :: T$=T$&SEG
$(M$(N),3,1)
650 NEXT I
660 IF Z(X)>4 THEN 760
670 Q$=F$(Z(X)+2,X):: FOR I=
1 TO LEN(Q$)STEP 2 :: N=VAL(
SEG$(Q$,I,2)):: IF SEG$(M$(N
),3,1)="3" THEN 680 ELSE V(1
)=MAX(V(1),VI(N,1)):: V(2)=M
AX(V(2),VI(N,2))
680 NEXT I
690 IF Z(X)>3 THEN 720
700 Q$=F$(Z(X)+3,X):: FOR I=
1 TO LEN(Q$)STEP 2 :: N=VAL(
SEG$(Q$,I,2)):: IF SEG$(M$(N
),3,1)="3" THEN 710 ELSE W(1
)=MAX(W(1),VI(N,1)):: W(2)=M
AX(W(2),VI(N,2))
710 NEXT I
720 IF V(1)=2 OR V(2)=2 THEN
 SU(X)=SU(X)/2
730 IF V((CHA-120)/8)=3 THEN
 SU(X)=.2
740 IF V((144-CHA)/8)=3 THEN
 SU(X)=.1
750 IF W((CHA-120)/8)=3 AND
V((CHA-120)/8)=3 AND V((144-
CHA)/8)<3 THEN 800
760 IF SU(X)>SUM THEN SUM=SU
(X):: X1=X
770 IF SU(X)=SUM AND RND<.5
THEN X1=X
780 NEXT X
790 X=X1
```

LISTINGS

```
800 DISPLAY AT(21,10):"MEIN
ZUG:";X :: Y,Z(X)=Z(X)+1 ::
CALL SETZEN(Y,X,CHA)
810 Q$=F$(Y,X)
820 FOR I=1 TO LEN(Q$)STEP 2
 :: N=VAL(SEG$(Q$,I,2))
830 VI(N,(CHA-120)/8)=VI(N,(
CHA-120)/8)+1
840 IF VI(N,(CHA-120)/8)=4 T
HEN DISPLAY AT(21,8):COL$((C
HA-120)/8);" HAT VIER" :: CA
LL VIER(M$(N),CHA):: RUN
850 NEXT I :: GOTO 410
860 FOR I=1 TO 2 :: DISPLAY
AT(21,1):"WIE HEISST DER ";S
TR$(I);". SPIELER ?"
870 ACCEPT AT(23,1)BEEP SIZE
(10):NAME$(I):: NEXT I :: CA
LL HCHAR(21,1,32,128)
880 IF ZUG>41 THEN DISPLAY A
T(21,9):"UNENTSCHIEDEN" :: R
UN ELSE CHA=264-CHA :: DISPL
AY AT(21,10)BEEP:NAME$((CHA-
120)/8);",": :TAB(10);"IHR Z
UG ?"
890 CALL KEY(0,K,S):: IF K<4
9 OR K>55 THEN 890
900 CALL HCHAR(23,21,K):: X=
K-48 :: IF Z(X)=6 THEN CALL
SOUND(100,220,0):: GOTO 890
910 Y,Z(X)=Z(X)+1 :: CALL SE
TZEN(Y,X,CHA):: ZUG=ZUG+1
920 Q$=F$(Y,X):: FOR I=1 TO
LEN(Q$)STEP 2 :: N=VAL(SEG$(
Q$,I,2)):: VI(N,(CHA-120)/8)
=VI(N,(CHA-120)/8)+1
930 IF VI(N,(CHA-120)/8)<4 T
HEN 950 ELSE DISPLAY AT(21,8
):COL$((CHA-120)/8);" HAT VI
ER": :TAB((16-LEN(NAME$((CHA
-120)/8)))/2);NAME$((CHA-120
)/8);" HAT GEWONNEN"
940 CALL VIER(M$(N),CHA):: R
UN
950 NEXT I
960 GOTO 880
970 SUB SETZEN(Y,X,CHA)
980 CALL SPRITE(#1,96,12-(CH
A-128)/4,1,57,0,20)
990 CALL COINC(#1,1,(7+2*X)*
8+1,5,C):: IF C THEN CALL MO
TION(#1,0,0)ELSE CALL SOUND(
10,-1,0):: GOTO 990
1000 CALL LOCATE(#1,1,(7+2*X
)*8+1):: FOR J=1 TO 7-Y :: C
ALL LOCATE(#1,(2*J+1)*8+1,(7
+2*X)*8+1):: CALL SOUND(100,
440-J*10,5):: NEXT J
1010 CALL HCHAR(2*J,8+2*X,CH
A):: CALL HCHAR(1+2*J,8+2*X,
CHA+1):: CALL HCHAR(2*J,9+2*
X,CHA+2):: CALL HCHAR(1+2*J,
9+2*X,CHA+3)
```

LISTINGS

```
1020 CALL DELSPRITE(#1):: SU
BEND
1030 SUB VIER(A$,CHA)
1040 Y=ASC(A$)-48 :: X=VAL(S
EG$(A$,2,1))
1050 A=VAL(SEG$(A$,3,1))
1060 ON A GOTO 1070,1100,113
0,1160
1070 FOR I=0 TO 3
1080 CALL SPRITE(#I+1,96,12-
(CHA-128)/4,121-16*Y-16*I,57
+16*X+16*I)
1090 NEXT I :: GOTO 1190
1100 FOR I=0 TO 3
1110 CALL SPRITE(#I+1,96,12-
(CHA-128)/4,121-16*Y-16*I,57
+16*X-16*I)
1120 NEXT I :: GOTO 1190
1130 FOR I=0 TO 3
1140 CALL SPRITE(#I+1,96,12-
(CHA-128)/4,121-16*Y-16*I,57
+16*X)
1150 NEXT I :: GOTO 1190
1160 FOR I=0 TO 3
1170 CALL SPRITE(#I+1,96,12-
(CHA-128)/4,121-16*Y,57+16*X
+16*I)
1180 NEXT I
1190 F=12-(CHA-128)/4
1200 CALL KEY(0,K,S):: CALL
COLOR(#1,2,#2,2,#3,2,#4,2,#1
,F,#2,F,#3,F,#4,F):: IF S=0
THEN 1190
1210 CALL DELSPRITE(ALL)
1220 CALL SOUND(100,440,0)::
 SUBEND
1230 SUB TITEL
1240 CALL CHAR(64,"7C829AA2A
29A927C")
1250 CALL CHAR(96,"0F1020478
F9C989898989F98503F1F0FF0080
8F8F00000003F27C70F1FFFFBF7"
)
1260 CALL CHAR(100,"1F101717
1616171417171616171417F1FF808
F8F8000E020E0E00000F808F8F8
")
1270 CALL CHAR(104,"F3B4B5B5
B5B95B5B5B5B5B2F2E2E2E3ECFE9
EBEBEBF37636B6B6B69C5C5C5C7C
")
1280 CALL CHAR(108,"0302020202
020202020202020303C040
C0C0C0C0C0C0C0C0C0C0C0C0C0C0
")
1290 CALL CHAR(112,"3C222E2F
2F2D2D2C2E2E2D2D2C2C3C3C3C24
2C2C2CACACECEC6C6C3CBC9C5C3C
")
1300 CALL CHAR(116,"3F202F3E
0202020202020202020303FC0C
FCFCC0C0C0C0C0C0C0C0C0C0C0C0
")
1310 CALL CHAR(120,"3C2C2626
1613131309090909040405033C24
4C4C5898989B0B0B0B0E0E0E0C0
")
1320 CALL CHAR(124,"3F202F2F
2C2C2C2F2C2C2F2F2F2C3C3CF80C
FEE6262626C60E1CF8F0984C261E
")
1330 CALL CHAR(136,"0003F1F
3F3F7F7F7F7F3F3F1F0F030000C0
F0F8FCFCFEFEFEFEFCFCF8F0C000
")
1340 FOR I=5 TO 12 :: CALL C
OLOR(I,2,16):: NEXT I
1350 CALL SCREEN(16):: DISPL
AY AT(8,12)ERASE ALL:"xzlndf
|~":TAB(12)|"y{moeg)"|CHR$(1
27)
1360 DISPLAY AT(12,9):"'bdfh
jlnprprtv":TAB(9)|"acegikmoq
sqsuw"
1370 DISPLAY AT(20,8):"@ JEN
S TIEDEMANN"
1380 FOR I=1 TO 4 :: CALL SP
RITE(#I,136,10,123,64+25*I):
: NEXT I :: FA=10
1390 FA=22-FA :: CALL KEY(0,
K,S):: CALL COLOR(#1,FA,#2,F
A,#3,FA,#4,FA):: IF S THEN S
UBEXIT ELSE 1390
1400 SUBEND
```

BÖRSE

Verk.: TI99/4A, 2 Rec. Kabel, Rec., ca. 40 Spiele + Progr. Sehr viele Listings, Joystick, Bücher: TI Progr. (Spiele, Grafik, usw.). Tips + Tricks, Spielen/ Lernen/Arbeiten, Handb. Writing Work book, TIBasic/ Ex.Basic f. Anfänger.
Tel. 05257/2776 (Joachim Hubertus anfr.)

Verkaufe TI99/4A mit den Modulen Ext.-Bas., Datenvv. + Anal., Stat., Schach, Parsec, Munch Man, TI-Inv., Chish. Trail, Joystick m. Adapter, 25 Zeitschr. 3 Bücher, 350 Progr., Cass. Rec. m. Kabel zus. 400 DM.T.02821/48993

Verk.: Disk-Contr. 249 DM; Disk-Man. 50 DM; Joyst. 30 DM; XB 175 DM, Attack, Invaders, Chis. Trail je 25 DM; Soccer, Parsec je 30 DM; Con. Four 20 DM; Beg. Gram. 15 DM; Finanzber. 15 DM; Basicprogramm-Rout. 25 DM. Te. 0911/498713, 18h

---

Verkaufe TI-99/4A + XBasic + Miniass. + Assemblerbuch + Rec. Kabel + Monitor, alles für 490 DM. Tel. 02104/46184

Verkaufe: Videogames 1 + Attack (55 DM), Diskmanager 2 (40 DM), TI-Joysticks (40 DM).- Kaufe Schachmodul (45 DM) oder tausche gegen Videogames 1 + Attack. Verkauf per NN. Kai Breuer, Zur Burg 30, 6393 Wehrheim 1

Verk. TI-99/4A (80 DM), Ex-Basic (110 DM), RS232 + V24-Kabel (160), Speech-Syntehsizer (100 DM VB), Rec.-Kabel (10 DM), Joyst.-Adapt. (10 DM), 2 Bücher (35 DM), Munchman (29 DM), alles ohne Defekt
Tel. 06074/29556, bitte Daniel verlangen

Zu Verk.: 2 x P-Box, 2x RS232, 3x 32K, 4x Joyst., 4x Rec.-Kabel, 1x Sprachsynth 1400 Progr./180 Disks, Ex-B, E/A, Mini-Mem, Modem, TI-Writer, viele Module, viele Bücher, viele Extras. Tel. 06204/1471 (Piepton abwarten, Name, Nr. und Ang. geben.

Tausche TI-Revue Jahrg. 1985 (6 Hefte) gegen Sotware. Ang. bitte an Peter Voigt
Tel. 02162/56520 ab. 19.oo Uhr

LISTING

BRD-QUIZ

BRD-QUIZ ist ein Spiel- und Lernprogramm für die ganze Familie, bei dem jeder sein Wissen über die BRD beweisen kann, oder auch nicht . . .
Am Anfang spielt der Computer die Nationalhymne. Nachdem er dann die Mitspielerzahl (bis 4) und die Namen der Spieler abgefragt hat, kann die erste Runde beginnen. Der Computer nennt eine Stadt und eine Auswahl von vier Flüssen. Die Spieler sollen nun die Nummer des Flusses eingeben, der durch diese Stadt fließt, bzw. an dem die Stadt liegt. Sollten mehrere Flüsse durch die Stadt fließen, so gilt immer der längere. Dieses muß jeder dreimal machen. In der nächsten Runde zeichnet der Computer auf der BRD-Karte eine Stadt ein und gibt vier Städtenamen. Die Spieler sollen nun die Nummer der Stadt eingeben, die auf der Karte abgebildet ist. Dieses Wissen muß jeder dreimal unter Beweis stellen. In der vierten Runde geht es darum, Besonderheiten und charakteristische Merkmale von Städten zu kennen. Wiederum gibt der Computer eine Stadt vor und die Spieler müssen durch Drücken einer Taste von 1 – 4 die Besonderheit der Stadt aus vier Möglichkeiten eingeben. Dies muß zweimal gemacht werden. In der vierten Runde geht es wieder darum, den Fluß zu kennen, der durch eine bestimmte Stadt fließt, bzw. an dem die Stadt liegt. Nur diesmal muß jeder den Flußnamen selbst eingeben. Wieder gilt bei einer Stadt, durch die mehrere Flüsse fließen, immer der längere. Dies muß jeder dreimal spielen. In der nächsten Runde zeichnet der Computer auf der Karte Flüsse ein, deren Namen jeder Spieler selbst eingeben muß. Dieses muß jeder zweimal spielen. In der letzten Runde geht es wieder um Besonderheiten und charakteristische Merkmale von Städten. Der Computer gibt eine Besonderheit vor und die Spieler müssen die dazugehörige Stadt selbst eingeben. Falsche Antworten werden immer korrigiert; richtige werden mit einem Punkt belohnt. Zum Schluß schreibt der Computer die Punkte jedes Spielers auf den Bildschirm.
So kann jeder feststellen, ob er nicht doch besser in der Schule aufgepaßt hätte.

Zur Beachtung bei der Eingabe der Lösung:
Falls Umlaute bei der Eingabe vorkommen, müssen diese unbedingt eingegeben werden, und nicht z.B. Ä durch AE ersetzt werden.

Ä:	FCTN	R
Ö:	FCTN	Z
Ü:	FCTN	T
ß:	SHIFT	6

Mit FCTN S kann man mit dem Cursor nach links fahren und so korrigeren. Mit ENTER nimmt der Computer die Eingabe an. Es ist aber ratsam, die Eingabe vorher auf Schreibfehler hin zu überprüfen, da der Computer nur korrekt geschriebene Eingaben als richtig erkennt.

Zur Eingabe des Programms:
Das Programm wird wie abgedruckt eingegeben. Lediglich bei den Zeilen 1320 und 1330 ist zu beachten, daß ab CHR$(127) die Großbuchstaben und Zeichen zusammen mit der CTRL-Taste eingegeben werden müssen, also CTRL und abgedruckter Großbuchstabe bzw. Zeichen.

LISTINGS

```
10 REM*********************
11 REM*                     *
12 REM*     BRD-QUIZ        *
13 REM*  Copyright  by      *
14 REM*Wolfgang  Hesseler*
15 REM* Neustadterstr.6    *
16 REM* 5461 Nd.-Buchenau*
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                     *
23 REM* Speicherbelegung *
24 REM*   13124 Bytes       *
25 REM*********************
26 REM
160 CALL CLEAR
170 CALL SCREEN(15)
180 FOR I=2 TO 7
190 CALL COLOR(I,1,1)
200 NEXT I
210 PRINT "      ((((((((((((
((((((((         ((((((((((((
((((((((         ((((((((((((
((((((((         ((((((((((((
((((((((         ((((((((((((
(((((((("
220 PRINT "      ØØØØØØØØØØØØ
ØØØØØØØØ         ØØØØØØØØØØØØ
ØØØØØØØØ         ØØØØØØØØØØØØ
ØØØØØØØØ         ØØØØØØØØØØØØ
ØØØØØØØØ         ØØØØØØØØØØØØ
ØØØØØØØØ"
230 PRINT "      888888888888
88888888         888888888888
88888888         88888888
888888888888         88888888
888888888888         88888888
888888888888"
240 PRINT :"           BUNDESRE
PUBLIK":"           DEUTSCHLAN
D": : :
250 CALL COLOR(2,2,2)
260 CALL COLOR(3,7,7)
270 CALL COLOR(4,12,12)
280 FOR I=5 TO 7
290 CALL COLOR(I,2,1)
300 NEXT I
310 DATA 4400384444 7C4444,3,
311,44007C444444447C,1,349,4
400444444444438,2,392,382424
382424382,2,349
320 DATA ØØØØØØØØØØØØ6Ø6,2,4
15,ØØ1Ø78F8FCFC5,2,392,FFFFF
FFFFFFFFFFF,1,349,Ø040F8FDFF
FF7F3F,1,294
330 DATA ØØØØØØ8ØFØF8FCF8,2,
311,ØØØØØØØ1Ø3Ø3,2,523,3F1F1
FFFFFFF3F7F,2,466,F6CEFEFEFF
FFFFFF,2,415
340 DATA ØØØØØØ7ØF9FFFFFF,2,
392,ØØØØØØØØFØEØEØE,2,349,ØØ
ØØØØØØØØØØØØØ3,1,392,ØØØØØØØ
ØØØØØØ3C3,1,311
350 DATA 7F1F7F3F1FFFFFFF,4,
466,FFFFFEFEFFFFFFFF,3,311,C
ØØØØØØØØØØØØØ8,1,349,ØØØØØ3Ø
1Ø3Ø3Ø1,2,392
360 DATA 77FFFFFFFFFFF7F,2,
349,C7C3C9DDCFDFFFFF,2,415,F
FFFFFFFFEFCF8FE,2,392,808,1,
349,ØØØØØØØ1Ø1Ø1Ø3Ø3,1,294
370 DATA ØØ8ØFØFEFFFFFEF8,2,
311,Ø3Ø3Ø31F3F3F3F1F,2,523,Ø
ØØØØØØ8ØCØCØCØC,2,466,Ø3Ø3Ø7Ø
F1F3F1F1F,2,415
380 DATA FFFFFFFFFFFFFCF8,2,
392,EØCØCØEØEØC,2,349,ØØ317F
7F3F3F1F1F,1,392,3FFFFFFFFFF
FFFFF,1,311,F8F8F8F8FCFCECC
390 DATA 4,466,1F1F1F3F3F3F3
FFF,2,349,FFFFFEFEFEFFFFFF,2
,392,CØØØØØØØØØØØØ8,1,349,FF7
F7FFFFFFF7F3F,1,294
400 DATA FFFFFEFCFEFCFCFF,2,
233,3F3F1F1F1FØF1F3F,2,415,Ø
ØØØCØCØEØF1FDFC,2,392,ØØØØØØ
Ø4ØECFF7FF,1,349
410 DATA ØØØØØØØØ7CFEFFFF,1,
294,7F7F7F7F3F1F1FØ7,2,233,8
Ø8Ø8ØEØFØF8FØF,2,466,ØF1F1F1
F1FØFØ7Ø7,2,415
420 DATA EØEØFØF8F8FCFCFF,3,
392,Ø3Ø3Ø1,1,392,FFFFBF1FØC,
2,440,FFFFFFFF7F3701,1,440,C
ØEØFØF8FCFEFFFF,1,466
430 DATA ØØØØØØØØØØØ8ØC,4,4
66,ØØØ1Ø3Ø7ØF1F1F3F,3,622,EØ
FØFØFØFØF82,1,587,3F3F3F7F7F
7F7FFF,1,587,FFFFFEFCEØCØ8Ø8
440 DATA 1,523,ØØØØØØØØ1Ø1Ø
1Ø1,2,466,FFFFFFFFFFF3E1F,3,
523,FFFFFFFFFFFFFF8D,1,466,C
ØCØEØEØEØCØF8F8,1,466
450 DATA BF36,1,415,FØ2,2,39
2,19,3,349,FFEFØ3Ø1Ø1ØØØ1,.8
,392,FFEFEØFØEØEØC,.8,415,FF
FF7F7F3,1,466,FC8Ø8,1,523,81
460 DATA 1,415,FØFØ7,1,349,2
,311,1,392,1,349,4,466,3,622
,1,587,1,587,1,523,2,466,3,5
23,1,466,1,466,1,415
470 DATA 2,392,3,349,.5,392,
.5,415,1,466,1,523,1,415,1,3
49,2,311,1,392,1,349,4,311
480 FOR I=91 TO 159
490 READ A$,A,B
500 CALL SOUND(A*300,B,Ø,B-3
,10,B+3,10)
510 CALL CHAR(I,A$)
520 NEXT I
530 FOR I=1 TO 25
540 READ A,B
550 CALL SOUND(A*300,B,Ø,B-3
,10,B+3,10)
560 NEXT I
570 CALL CLEAR
580 CALL SCREEN(8)
```

LISTINGS

```
590 OPTION BASE 1
600 DIM N$(4),N(4),M$(4)
610 DATA 42,00003F3F0C0C0C0C
,44,0000F0F00C0C0C0C,46,0F0F
0C0C0C0C3F3F,47,F0F00C0C0C0C
F8F0,33,000406FFFF0604
620 DATA 35,0001F3F303030300
,36,0000C0E030303030,38,3F3F
333331303030,39,E0C0008C0E0
7038,64,007E7E7E7E7E7E7E
630 DATA 59,0003F3F1C0C0C0C
,60,0000C0E030303030,61,0C0C
0C0C0C1C3F3F,62,3030303030300
E0C0
640 FOR I=1 TO 14
650 READ A,A$
660 CALL CHAR(A,A$)
670 NEXT I
680 GOSUB 700
690 GOTO 880
700 FOR I=1 TO 4
710 CALL COLOR(I,1,1)
720 NEXT I
730 PRINT "    *,":"    ./":
 :"       ##":"       &'": :
"         ¦<":"         =>
": : : : : : : : : : :
740 CALL COLOR(3,2,1)
750 CALL COLOR(2,2,1)
760 CALL SOUND(45,-5,0)
770 CALL COLOR(1,7,1)
780 CALL SOUND(45,-5,0)
790 CALL COLOR(4,11,1)
800 CALL SOUND(45,-5,0)
810 FOR I=1 TO 5
820 CALL HCHAR(8,11+I*2,ASC(
SEG$("-QUIZ",I,1)))
830 CALL SOUND(45,-6,0)
840 NEXT I
850 FOR I=1 TO 100
860 NEXT I
870 RETURN
880 S$="1805WIEVIELE"
890 GOSUB 3200
900 S$="1905MITSPIELER"
910 GOSUB 3200
920 CALL SOUND(100,880,0)
930 F=24
940 CALL HCHAR(19,17,F+55)
950 FOR I=1 TO 10
960 CALL KEY(0,K,S)
970 IF (K<49)+(K>52)=0 THEN
1010
980 NEXT I
990 F=-F
1000 GOTO 940
1010 CALL SOUND(100,1000,0)
1020 CALL HCHAR(19,17,K)
1030 P=K-48
1040 IF A1 THEN 2900
1050 O=P
1060 FOR K=1 TO P
1070 CALL CLEAR
1080 GOSUB 700
1090 PRINT "NAME DES ";STR$(
K);"_SPIELERS": :
1100 CALL SOUND(200,440,0)
1110 CALL SOUND(200,349,0)
1120 INPUT "!":N$(K)
1130 NEXT K
1140 CALL CLEAR
1150 CALL SCREEN(11)
1160 FOR I=1 TO 8
1170 CALL COLOR(I,16,1)
1180 NEXT I
1190 S$="0504FANGEN WIR AN__
_"
1200 GOSUB 3200
1210 RANDOMIZE
1220 DEF R(X)=INT(X*RND)+1
1230 FOR I=1 TO 200
1240 NEXT I
1250 G=16
1260 CALL COLOR(4,7,11)
1270 FOR I=9 TO 16
1280 CALL COLOR(I,11,1)
1290 NEXT I
1300 CALL CLEAR
1310 CALL SCREEN(11)
1320 PRINT "    bc":"   defg
h":"  ijkalm":" nopaaqr":" s
aaaaat":" uaaaaav  '":" waaa
axy":"z{aaaa|":"}aaaa~"¦CHR$
(DEL):" aaaa ":" aaaaa   "
1330 PRINT " aaaaaaaa ":" aa
aaaaaa ":"   aaaaaaa  ":"
aaaaaaaa ":"   aaaaaaa ":"
a aaaaa  ":"             ": :
:
1340 CALL SCREEN(5)
1350 IF A1 THEN 2520 ELSE 32
60
1360 DATA LAHN,ELBE,ILMENAU,
ALLER,WESER,LEINE,WESER,INNE
RSTE,LEINE,"",EMSCHER,HASE,"
",FULDA,RHEIN,RHEIN,WUPPER
1370 DATA WUPPER,WUPPER,RHEI
N,RHEIN,SIEG,RHEIN,MOSEL,LAH
N,FULDA,RHEIN,RHEIN,MAIN,MAI
N,"",MAIN,RHEIN,RHEIN,RHEIN
1380 DATA ISENACH,RHEIN,MURG
,"",NECKAR,MAIN,MAIN,PEGNITZ
,DREISAM,DONAU,ISAR,ISAR,LEC
H,DONAU,NECKAR,NECKAR,ENZ
1390 RESTORE 1360
1400 A=R(52)
1410 FOR I=1 TO A
1420 READ A$
1430 NEXT I
1440 IF A$="" THEN 1390
1450 RESTORE 1760
1460 FOR I=1 TO A
1470 READ B$
1480 NEXT I
1490 B$=SEG$(B$,8,12)
1500 IF A1<6 THEN 1540
```

```
1510 C$=A$
1520 A$=B$
1530 B$=C$
1540 IF A1>3 THEN 2530
1550 L=R(4)
1560 M$(L)=A$
1570 FOR I=1 TO 3
1580 A=R(52)
1590 IF A1=3 THEN 1620
1600 RESTORE 1360
1610 GOTO 1630
1620 RESTORE 2080
1630 FOR J=1 TO A
1640 READ C$
1650 NEXT J
1660 FOR J=1 TO 4
1670 IF M$(J)=C$ THEN 1580
1680 NEXT J
1690 FOR J=1 TO 4
1700 IF M$(J)<>"" THEN 1730
1710 M$(J)=C$
1720 GOTO 1740
1730 NEXT J
1740 NEXT I
1750 GOTO 2790
1760 DATA 1081305MARBURG,072
0706HAMBURG,0220807L]NEBURG,
1340906CELLE,0640805BREMEN,0
211006HANNOVER,0911005HAMELN
1770 DATA 0921006HILDESHEIM,
1541106G\TTINGEN,0021103M]NS
TER,1441103DORTMUND,0441004O
SNABR]CK,1341004BIELEFELD
1780 DATA 0911205KASSEL,0241
202DUISBURG,0841202D]SSELDOR
F,1181202WUPPERTAL,1211202RE
MSCHEID,1221202SOLINGEN
1790 DATA 0441302K\LN,138130
2BONN,0481304SIEGEN,1021403K
OBLENZ,1481502TRIER,0381404W
ETZLAR,0441406FULDA
1800 DATA 0511503WIESBADEN,0
781503BINGEN,0321504FRANKFUR
T,0481505OFFENBACH,1541504DA
RMSTADT,0221505HANAU
1810 DATA 0711503MAINZ,04216
04WORMS,1211604SPEYER,114160
3BAD D]RKHEIM,1211704KARLSRU
HE,0281804RASTATT
1820 DATA 0681804BADEN-BADEN
,1541604HEIDELBERG,0121606W]
RZBURG,1081509BAYREUTH,14116
08N]RNBERG,0212003FREIBURG
1830 DATA 1111811PASSAU,0481
810LANDSHUT,1441909M]NCHEN,0
441908AUGSBURG,0141906ULM,05
81805STUTTGART,0721805ESSLIN
GEN,0141804PFORZHEIM
1840 RESTORE 1760
1850 GOSUB 3160
1860 W=0
1870 FOR I=1 TO 4
1880 A=R(13)
1890 FOR J=1 TO 13
1900 READ A$
1910 IF J=A THEN 1980
1920 NEXT J
1930 NEXT I
1940 IF W=0 THEN 1840
1950 GOSUB 2830
1960 CALL HCHAR(VAL(SEG$(C$,
4,2)),VAL(SEG$(C$,6,2))+2,97
)
1970 RETURN
1980 B=R(4)
1990 IF M$(B)<>"" THEN 1980
2000 M$(B)=SEG$(A$,8,12)
2010 IF (RND>I*.17)+(W=1)THE
N 1920
2020 W=1
2030 L=B
2040 C$=A$
2050 CALL CHAR(61,SEG$("0000
0000000000",1,J)&SEG$(A$,3,
1))
2060 CALL HCHAR(VAL(SEG$(A$,
4,2)),VAL(SEG$(A$,6,2))+2,61
)
2070 GOTO 1920
2080 DATA GRABMAHL D_HL_  EL
ISABETH,WICHTIGSTE DT_  SEEH
AFENSTADT,NAMENSGEBER D_  GR
\^TEN DT_HEIDE
2090 DATA GR\^TER EUROP_  OR
CHIDEENBETRIEB,STADTMUSIKANT
EN,HAUPTSTADT V_  NIEDERSAC
HSEN,RATTENF]NGER
2100 DATA 1000J]HRIGER  RO
SENSTOCK,MAX-PLANCK-GE-  SEL
LSCHAFT(SITZ),WESTF]LISCHER
 FRIEDE M_FRANKR_
2110 DATA WESTFALENHALLE,WES
TF]HL_FRIEDE M_SCHWEDEN,DEUT
SCHES SPIEL-KARTENMUSEUM,BUN
DESSOZIAL-  GERICHT,GR\^TER
 BINNEN- HAFEN D_WELT
2120 DATA HAUPTSTADT V_  NR
W,SCHWEBEBAHN,ZENTRUM D_DT_
 WERKZEUGIND_,KLINGEN+SCHNE
IDEWARENZENTRUM,GR\^TES DT_G
OT_ BAUWERK(DOM),HPTST_D_BRD
2130 DATA GEBURTSORT PETERPA
UL RUBENS,BUNDESARCHIV,]LTES
TE DT_STADT,REICHSKAMMERGE-
RICHT 1693-1806
2140 DATA DOM M_GRAB D_  HL
_BONIFATIUS,BUNDESKRIMINAL-
AMT,M]USETURM,GR\^TER DT_FLU
G-HAFEN
2150 DATA HAUPTSITZ D_ DT_LE
DERWARENIND_,DT_RECHENZENTRU
M,GEBURTSORT D_  GEBR_GRIMM
,SITZ D_ZDF
2160 DATA REICHSACHT ]BER LU
THER 1521,KAISERDOM M_GR]-BE
RN DT_KAISER,RIESENWEINFA^,B
```

```
UNDESVERFAS-   SUNGSGERICHT
2170 DATA BEENDUNG D_SPAN_ER
BFOLGEKRIEGES,HEISSESTE EURO
P_THERMALQUELLE,[LTESTE UNIV
ER- SITI[T D_BRD
2180 DATA FESTUNG          MA
RIENBERG,STADT FRANZ      LIS
ZTS,GERMANISCHES    NATIONAL
MUSEUM,[LTESTER DT_GAST-HOF
2190 DATA GR\^TE KIRCHEN- OR
GEL,F]RSTENHOCHZEIT,DEUTSCHE
S MUSEUM,FUGGER,H\CHSTER KIR
CH- TURM D_WELT
2200 DATA HAUPTSTADT V_    BA
DEN-W]RTTEMB_,GR]NDUNG D_SCH
W[-BISCHEN BUNDES,ZENTRUM D_
DT_    SCHMUCKWARENIND_
2210 RESTORE 2080
2220 CALL COLOR(2,16,1)
2230 CALL COLOR(4,16,1)
2240 GOTO 1400
2250 DATA #,ELBE,4,FFFFFFFFF
E3CC0FF,7,7,BFDFE7FBFDFEFFFF
,7,6,FFFFFFFFFFFFFF7F,6,6,7F
1F7F3F1F81FEFF,6,5,#,ALLER,4
2260 DATA 00C0C0E0E0C,10,8,F
FFFFFFF7F9FEFF,9,7,7F9FEFF1F
EFFFFFF,9,6,FEFFFFFFFFFFFFFF
,9,5,#,EMS,5,1FEFEDF3FFFFFFF
F
2270 DATA 11,4,7F7FBFCFEFEFF
3F8,10,3,7FFFFFFFFFFFF7F,9,
3,0302021E3E3E3E1F,9,2,BFBFB
F7FBFBF7F7F,8,3,#,WESER,7
2280 DATA 7F7F7FFFFFF3FDF,1
1,6,FFFFFFEFEFEFEFF,11,5,F7
F7EFDFC3FDFEFE,10,5,FDFDFBFD
FDFBFBF7,9,5,FFFF7FBFDFE7FBF
D
2290 DATA 8,5,C7C3C9DDCDDDFD
FD,7,4,FDFEFFFFFFFFFFFF,8,4,
#,FULDA,3,FFFF9EBEBE3F7FBF,1
2,6,9FEFF6F4EEDCBCBF,13,6
2300 DATA BFBFDFEFF1FFFFFF,1
4,6,#,LIPPE,4,FFFFFFDF3FFFFF
FF,11,5,FFFFFFFFFEF807FF,11,
4,FFFFFFFFFF7986FF,11,3
2310 DATA 3F7FFFFFFFF807FF,1
1,2,#,RUHR,3,FFFF7F87FBFBFBF
F,12,4,FFF24DBFFFFFFFFF,12,3
,FFBFC0FFFFFFFFFF,12,2
2320 DATA #,LAHN,4,FFFFF7FBF
CFFFFFE,13,4,FFFFFFFF7F7F7FF
F,13,5,FEF9C7DF1FFFFFFF,14,4
,FFFFFFFFFFFCF1FF,14,3
2330 DATA #,MOSEL,3,001F1F1F
1F0F0707,16,1,FDF9FBF7EFEF9F
7F,15,2,FFFFFFFFFFFFFFFC,14,
2
2340 DATA #,INN,3,BF7F7F7F7F
7F7F8D,20,10,FFFFFFF8F7CFDFB
F,19,10,FFFFFE7C8080000,19,1
1
2350 DATA #,ISAR,6,FFFE7D7D3
,21,8,BF,21,9,DFBFBFDFDFDFDF
DF,20,9,FEF9F7F7EFDFDFDF,19,
9,FFFFFFFFFEF1CF3F,18,10
2360 DATA FFFFFFBF7FFFFFFF,1
8,11,#,LECH,4,FEEDE0F0E0E0C,
21,7,BFBF9FDFDFBFBF7F,20,8,B
FBFBFBFDFDFBFBF,19,8
2370 DATA FFFFFFFFDFDFDFBF,1
8,8,#,NECKAR,5,FEFEFDFDFDFDF
BFB,19,4,FBFBFDFDFEF9F70F,18
,5,F7F7F7FBFBF7F7FB,17,5
2380 DATA FFFFFFFFFFFF9F6F,1
6,5,FFFFFFFFFBFDFDFE,16,4,#,
WERRA,1,FFFFDEE6FAFBFDFE,12,
6
2390 CALL COLOR(1,11,1)
2400 RESTORE 2250
2410 FOR I=1 TO R(14)
2420 READ B$,A$,C
2430 IF B$<>"#" THEN 2420
2440 NEXT I
2450 FOR I=1 TO C
2460 READ B$,A,B
2470 CALL CHAR(I+32,B$)
2480 CALL HCHAR(A,B+2,I+32)
2490 NEXT I
2500 GOSUB 2540
2510 GOTO 1300
2520 RETURN
2530 GOSUB 2790
2540 R$=""
2550 H=17
2560 CALL SOUND(100,880,0)
2570 CALL KEY(0,K,S)
2580 IF S=1 THEN 2620
2590 G=-G
2600 CALL HCHAR(13,H,G+48)
2610 GOTO 2570
2620 IF K=13 THEN 2730
2630 IF K<>8 THEN 2680
2640 IF H<18 THEN 2570
2650 CALL HCHAR(13,H,32)
2660 H=H-1
2670 GOTO 2590
2680 IF (K<>32)*(K<>45)*(K<6
5)+(K>94)+(H=29)THEN 2570
2690 CALL HCHAR(13,H,K)
2700 H=H+1
2710 CALL SOUND(10,300,10)
2720 GOTO 2570
2730 CALL HCHAR(13,H,32)
2740 FOR I=17 TO 30
2750 CALL GCHAR(13,I,A)
2760 R$=R$&CHR$(A)
2770 NEXT I
2780 IF SEG$(R$,1,POS(R$,"
",1)-1)=A$ THEN 3090 ELSE 29
10
2790 FOR I=1 TO LEN(B$)
2800 CALL HCHAR(8-(I>16),16+
I+16*(I>16),ASC(SEG$(B$,I,1)
```

```
))
2810 NEXT I
2820 IF A1>3 THEN 3190
2830 FOR H=1 TO 4
2840 FOR J=0 TO INT(LEN(M$(H
))/17)
2850 S$=STR$(H*2+8+J)&STR$(1
6+(A1+J=3)*2)&SEG$(STR$(H)&"
)"&M$(H),J*18+1,18)
2860 GOSUB 3200
2870 NEXT J
2880 NEXT H
2890 GOTO 920
2900 IF L=P THEN 3090
2910 FOR I=0 TO 30
2920 CALL SOUND(-37,260-I*5,
I)
2930 NEXT I
2940 FOR J=1 TO 3
2950 CALL SOUND(50,-2,0)
2960 IF A1>3 THEN 3020
2970 CALL HCHAR(L*2+8,17+(A1
=3)*2,33)
2980 FOR I=1 TO 25
2990 NEXT I
3000 CALL HCHAR(L*2+8,17+(A1
=3)*2,L+48)
3010 GOTO 3050
3020 CALL HCHAR(13,17,32,13)
3030 S$="1316"&A$
3040 GOSUB 3200
3050 FOR I=1 TO 20
3060 NEXT I
3070 NEXT J
3080 GOTO 3140
3090 FOR I=6 TO 18 STEP 2
3100 CALL SOUND(-37,587+I*5,
I)
3110 CALL SOUND(-37,623+I*5,
I)
3120 NEXT I
3130 N(A4)=N(A4)+1
3140 CALL HCHAR(1,12,32,20)
3150 CALL VCHAR(1,15,32,432)
3160 FOR I=1 TO 4
3170 M$(I)=""
3180 NEXT I
3190 RETURN
3200 E=VAL(SEG$(S$,1,2))
3210 F=VAL(SEG$(S$,3,2))
3220 FOR I=1 TO LEN(S$)-4
3230 CALL HCHAR(E,F+I,ASC(SE
G$(S$,I+4,1)))
3240 NEXT I
3250 RETURN
3260 FOR A1=1 TO 6
3270 DATA 3,3,2,3,2,2
3280 RESTORE 3270
3290 FOR I=1 TO A1
3300 READ A2
3310 NEXT I
3320 FOR A3=1 TO A2
3330 FOR A4=1 TO 0
```

LISTINGS

```
3340 S$="0102RUNDE "&STR$(A1
)&"  SPIELER "&STR$(A4)&"  P
UNKTE "&STR$(N(A4))
3350 GOSUB 3200
3360 ON A1 GOSUB 1390,1840,2
210,1390,2390,2210
3370 NEXT A4
3380 NEXT A3
3390 NEXT A1
3400 CALL CLEAR
3410 S$="0308ENDE DES SPIELS
"
3420 GOSUB 3200
3430 CALL HCHAR(4,9,45,15)
3440 S$="0804ERREICHBARE PUN
KTZAHL:15"
3450 GOSUB 3200
3460 FOR J=1 TO 0
3470 S$=STR$(J*2+8)&"04"&STR
$(J)&")"&SEG$(N$(J)&"
    ",1,10)&": "&STR$(N(J))
3480 GOSUB 3200
3490 NEXT J
3500 CALL KEY(0,K,S)
3510 IF S=0 THEN 3500
3520 CALL CLEAR
```

ANMERKUNG DER REDAKTION:

Liebe TI-User,

Aufgrund der Programmlänge ist dieses Programm nur mit Kassettenrecorder ladbar. Auf Diskette würde es 56 Sektoren belegen und damit im Dis/Var-Format vom Rechner abgespeichert werden. Von TI-Basic aus ist dieses Format nicht ladbar und in Extended-Basic läuft es nicht, weil der Autor für seine Deutschlandkarte den gesamten Zeichensatz bis einschließlich ASCII 159 umdefiniert hat, von der Farbgebung ganz abgesehen. Trotzdem haben wir uns entschlossen, dieses Programm zu veröffentlichen, da es gegenüber den allseits bekannten Ballerspielen eine geradezu wohltuende Abwechslung darstellt.

Für Euch heißt es damit aber, tippen, tippen, tippen. Denn wie bereits erklärt, ist dieses Programm in unserem Diskettenservice aus naheliegenden Gründen nicht erhältlich.

Bis auf die Zeilen 1320 und 1330 gibt es auch keine Schwierigkeiten beim Abtippen. In diesen beiden Zeilen hat der Autor allerdings bei der Programmierung reichlich mit der CTRL-Taste gearbeitet, um Teile seiner Deutschlandkarte zu definieren und es ist nicht ganz einfach, diese beiden Zeilen richtig einzugeben, obwohl der Autor den Eingabevorgang in seiner Dokumentation genau beschrieben hat.

Deshalb haben wir, in weiser Voraussicht daß trotzdem einige User an diesen beiden Zeilen kapitulieren müssen, in unserem Kopfbild die volle Anschrift des Autors eingefügt.

Wendet Euch bei Problemen also bitte direkt an den Verfasser dieses Programmes und vergeßt dabei nicht, Euerem Hilfeschrei einen frankierten Rückumschlag beizufügen.

Der Autor kann sein Honorar für dieses Programm schließlich nicht gleich wieder in Briefmarken umsetzen.

X-FORMAT 28

Viele Anfragen haben uns schon erreicht, ob wir unser Programm, mit dem wir die Listings im Bildschirmformat erstellen, nicht einmal abdrucken könnten.
Doch leider mußten wir diesem Wunsch jedesmal widersprechen, da auf unserer Version des Programmes ein Copyright liegt.
Viele Leser haben deshalb versucht, unser Programm nachzuschreiben. Doch die Ergebnisse waren bisher durchweg dürftig und nicht wert, sie Ihnen als Ersatz für unser eigenes Programm anzubieten. Bis uns das Listing und die Diskette des Herrn Weidmann erreichte.
Nachdem wir uns überzeugt hatten, daß dieses Programm nirgends ab- oder umgeschrieben wurde, entschlossen wir uns sofort, es allen TI-Usern zur Verfügung zu stellen. Zumal es von Herrn Weidmann fix und fertig zur Veröffentlichung bearbeitet wurde, einschließlich unseres Kopfbildes.
Neidlos müssen wir ihm zugestehen, daß seine Version sogar noch um einige Längen besser und komfortabler ist, als unsere eigene.
Nur eines hat der Autor nicht bedacht, was zu unverständlichen Ausdrucken führen kann:
Nicht jeder Drucker ist über die DIP-Schalter ständig auf den US-ASCII eingestellt. User, die nicht nur Probeausdrucke ihrer Listings machen, sondern auch nebenbei noch Texte schreiben, haben ihren Drucker sicher auf den deutschen Standard eingestellt. Im Programm "XFormat-28" wird dies jedoch nicht berücksichtigt und aus den Programmen würden sich alle nationalen Sonderzeichen beim Ausdruck in den deutschen Zeichensatz umwandeln, d.h., aus einer geschweiften Klammer nach links würde ein kleines "ä" entstehen, usw.
Deshalb muß jeder Interessent nach der Druckereröffnung noch zusätzlich die Codes für den US-Standard einfügen, damit sein Programm auch korrekt ausgedruckt wird.
In der Programmbeschreibung sind die benutzten Steuercodes sehr gut dokumentiert. Wir haben jedoch bewußt darauf verzichtet, einen zusätzlichen Code einzufügen, da ja bekanntlich nicht jeder von Euch den selben Drucker verwendet.
Nehmt also Euer Druckerhandbuch zu Hilfe und ergänzt das Listing um den einen Code.
Selbstverständlich muß auch die Druckereröffnung an Euere Schnittstelle angepaßt werden.
Danach besitzt Ihr ein hervorragendes Utility, das Euch die Kontrolle über Euere Programme wesentlich erleichtert und die Fehlersuche auf ein Minimum der Zeit reduziert.

PROGRAMMBESCHREIBUNG XFORMAT-28

Das Programm XFormat-28 erlaubt es, ein Basic-Programm im Bildschirmformat (28 Zeichen pro Zeile) zu drucken. Dabei werden immer 2 Spalten mit je 50 Zeilen auf eine Seite gedruckt. Die Zeilennummern werden durch Fettdruck hervorgehoben. Die Seiten werden fortlaufend numeriert. Eine eingegebene Kopfzeile wird auf jeder Seite

LISTINGS

```
100 ! ************************
110 ! *                      *
120 ! *     XFORMAT-28       *
130 ! *                      *
140 ! *    Copyright by      *
150 ! *                      *
160 ! *   Reiner Weidmann    *
170 ! *                      *
180 ! * Benoetigte Geraete   *
190 ! *  TI99/4A Konsole     *
200 ! *     Ext. Basic       *
210 ! *   DISK + Drucker     *
220 ! *                      *
230 ! *  Speicherbelegung    *
240 ! *     6025 Bytes       *
250 ! *                      *
260 ! ************************
270 !
280 DIM R$(100),P$(3)
290 GOTO 310 :: L,S,K,V,J,Z1
,Z2,I,M,C,T,A=P :: I$,X$,B$,
H$,A$=T$
300 !@P-
310 DISPLAY AT(8,5)ERASE ALL
:"WIEVIELE KOPIEN? 1"
320 ACCEPT AT(8,22)BEEP SIZE
(-2)VALIDATE(DIGIT):C
330 DISPLAY AT(10,1):"KOPFZE
ILE: (MAX. 50 ZEICHEN)"
340 LINPUT T$ :: T$=SEG$(T$,
1,50)
350 OPEN #1:"DSK1.LIST",INPU
T
360 OPEN #2:"PRT",VARIABLE 2
55 :: PRINT #2:CHR$(27);"@";
CHR$(27);"1";CHR$(10);CHR$(2
7);"8";
370 IF EOF(1)THEN 630 :: LIN
PUT #1:A$ :: H$=""
380 IF A$="" THEN 370 :: B$=
SEG$(A$,1,POS(A$," ",1)-1)
390 IF LEN(A$)=80 THEN 410
400 A$=A$&H$ :: IF LEN(H$)=8
0 THEN 410 :: GOSUB 480 :: G
OTO 370
410 IF EOF(1)THEN GOSUB 480
:: GOTO 630 ELSE LINPUT #1:H
$
420 P=POS(H$," ",1):: IF P<2
 OR P>6 THEN 400
430 ON ERROR 400
440 A=VAL(SEG$(H$,1,P-1))
450 IF A<1 OR A>32767 OR A<=
VAL(B$)THEN 400
460 GOSUB 480 :: A$=H$
470 IF LEN(A$)=80 THEN 410 E
LSE GOSUB 480 :: GOTO 370
480 P$(I)=A$ :: I=I+1 :: IF
I<3 THEN 530
490 Z1=VAL(SEG$(P$(1),1,POS(
P$(1)," ",1)-1))
500 Z2=VAL(SEG$(P$(2),1,POS(
P$(2)," ",1)-1))
510 IF Z1<Z2 THEN GOSUB 540
:: P$(0)=P$(1)ELSE P$(0)=P$(
0)&P$(1)
520 I=2 :: P$(1)=P$(2)
530 RETURN
540 P=POS(P$(J)," ",1):: X$=
CHR$(27)&"E"&SEG$(P$(J),1,P-
1)&CHR$(27)&"F"&SEG$(P$(J),P
,LEN(P$(J))-P+1):: V=33
550 R$(K)=SEG$(X$,1,32):: K=
K+1 :: IF K>99 THEN GOSUB 59
0
560 IF SEG$(X$,V,28)="" THEN
 580 :: R$(K)=SEG$(X$,V,28):
: V=V+28 :: K=K+1 :: IF K>99
 THEN GOSUB 590
570 GOTO 560
580 RETURN
590 L=INT((K-1)/2):: S=S+1 :
: FOR M=1 TO C :: INPUT "NEU
ES BLATT EINGELEGT ?":I$
600 PRINT #2:T$;TAB(54-(T=0)
*7);:: PRINT #2,USING "SEITE
: ##":S :: PRINT #2 :: T=1
610 FOR K=0 TO L :: PRINT #2
:R$(K);TAB(35-(SEG$(R$(K),1,
1)=CHR$(27))*4);R$(K+L+1)::
IF M=C THEN R$(K),R$(K+L+1)=
"" :: NEXT K ELSE NEXT K
620 PRINT #2:CHR$(12);CHR$(7
):: NEXT M :: K=0 :: RETURN
630 GOSUB 540 :: J=1 :: GOSU
B 540 :: IF K THEN GOSUB 590
640 CLOSE #1 :: CLOSE #2
```

wiederholt und dient der Identifizierung des Listings.
Um ein Programm mit XFormat-28 zu drucken, muß es vorher mit LIST "DSK1.LIST" auf Diskette gelistet werden. SFormat-28 liest nun jeweils 100 Zeilen zu 28 Zeichen in den Speicher und fordert dann den Bediener auf, ein neues Blatt in den Drucker einzulegen. Als Antwort genügt es, die ENTER-Taste zu drücken. Beim Drucken auf Endlospapier braucht natürlich kein neues Blatt eingelegt zu werden.
XFormat-28 läuft bei mir mit einem externen Interface der Fa. Radix ("PRT") und einem Epson RX 80 FT. Für andere Konfigurationen ist XFormat-28 ggf. anzupassen. Die verwendeten Funktionscodes des Druckers bedeuten:
CHR$(27);"§"
– Drucker auf Standardfunktionen setzen
CHR$(27);"1";CHRS(10)
– linker Rand auf Spalte 10
CHR$(27);"8"
– Papierendefühler "aus"
CHR$(27);"E"
– Fettdruck
CHR$(27);"F"
– Normaldruck
~~CHRS(7)~~
CHR$(12)
– Form Feed
CHR$(7)
– Beep

Reiner Weidmann

STUFEN ZUM ERFOLG

Dies ist ein Spiel aus der Serie der Hammurabi-Spiele, geschrieben für einen Spieler und einen Computer (natürlich einen TI99/4A).
Sinn und Zweck dieses Spieles ist, sich durch geschickte Transaktionen und eine den Gegebenheiten angepaßte Staatsführung vom Herrn bis zum Kaiser eines fiktiven Landes hochzuarbeiten.
Wann die nächste Stufe auf der Erfolgsleiter fällig ist, berechnet der Computer an Hand Ihrer Aktionen.
Um das gesetzte Ziel zu erreichen, müssen Sie günstig Land und Getreide einkaufen und wieder abstoßen, wenn es einen Gewinn erzielt. Natürlich müssen Sie dabei auch darauf achten, daß für Ihre Bevölkerung noch genügend Nahrungsmittel zum Leben und Land zum Bestellen übrig bleibt.
In jedem Jahr wirft das bestellte Land einen bestimmten Ertrag an Weizen ab, der von der Größe des Landes, der Bevölkerungsdichte, der Aussaat und dem Wetter abhängig ist.
Natürlich müssen Sie, als Schutzpatron Ihres Landes, auch die Bevölkerung zur Kasse bitten, indem Sie die Steuern festlegen. Aber Achtung, greifen Sie Ihren Schutzbefohlenen zu tief in die Tasche, wird sich unweigerlich auch die Auswanderungsrate erhöhen.
Zu einer gut florierenden Infrastruktur gehören natürlich auch Mühlen, die das geerntete Getreide weiterverarbeiten und selbstverständlich dürfen auch Fabriken nicht fehlen, wenn Sie nicht gerade einen Bauernstaat aufbauen wollen. Um Ihre Waren zu verkaufen, müssen Märkte angeschafft werden und zu einem representativen Regenten gehören natürlich auch Paläste.
Diese nicht gerade billigen Investitionen müssen erst einmal verdient werden.
Leider benötigt ein Staat auch Militär, um seine Grenzen gegen (eingebildete) Angriefer zu schützen. In diesem Spiel haben Sie die Möglichkeit, bis zur Großmacht aufzurüsten. Fußvolk, Artillerie, Kavallerie und verschiedene Seestreitkräfte können angeworben werden. Je mehr Militär Sie besitzen, desto höher steigt die Wahrscheinlichkeit, daß Sie aus einem eventuellen Krieg als Sieger hervorgehen und dadurch zusätzliche Gewinne erzielen.
Ihr einziger Wunsch gilt jedoch dem Traum, so schnell wie möglich Kaiser zu werden. Seien Sie ein gerechter Landesherr und ein gewitzter Geschäftsmann, dann dürfte diesem Ziel eigentlich nichts mehr im Wege stehen, oder . . . ???

Anmerkung der Redaktion:

Leider hat der Autor dieses interessante Spiel nur für eine Person und einen Computer geschrieben. Um wieviel könnte es gewinnen, wenn sich mehrere Personen im Wettstreit messen könnten. Wer wird zuerst Kaiser, wer ist der geschickteste Händler, wer der gerechteste Landesvater und wer der beste Kriegsherr?
Auch könnte man den Zuwachs an Märkten, Müh-

»

LISTINGS

```
10 ! ********************
11 ! *                  *
12 ! * STUFEN ZUM ERFOLG *
13 ! *     Vorprogramm  *
14 ! *    Copyright  by  *
15 ! *   Michael Haasler *
16 ! *                  *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole *
21 ! *     Ext. Basic   *
22 ! *                  *
26 ! *  Speicherbelegung *
27 ! *     2105 Bytes   *
28 ! *                  *
29 ! ********************
30 !
110 RANDOMIZE :: CALL MAGNIF
Y(3):: RESTORE :: CALL CLEAR
120 DATA 003F666C746466EF,00
7E313B3633337E,003E634140603
F1E,007E321119113F7E,007F333
03C38317F,007F723239383078
130 DATA 003E6648504F663E,00
7C603F3132327A03,007C38181C1
A183C2,003F13060606263F
140 DATA 007B72363C363673,00
7830303072637F
150 DATA 007F2B2A49494963,00
7C76323A323477,003E634559617
F3E,007E33313F3E3078,003E634
559617F3E07
160 DATA 007F313A34323179,00
3E413C06634F3E
170 DATA 007F492A0808081C,00
6F2626666E7E37,0073622232321
E0C,0077222241494936,0079321
40814264F
180 DATA 007341623418183C,00
7F43260C1A317F
200 FOR J=1 TO 160 :: NEXT J
 :: FOR J=65 TO 90 :: READ Z
$ :: CALL CHAR(J,Z$):: NEXT
J
241 CALL SCREEN(2):: FOR I=2
 TO 10 :: CALL COLOR(I,5,2):
: NEXT I :: CALL CHAR(120,"0
1030119254380809898804 03F3F3
F00008000304884020232320204F
8F8F8")
260 CALL CHAR(128,"000000001
030509112929457D39071F103068
A8447C380000000000000C0F0")
270 CALL CHAR(136,"00000080C
0FFFFFCFBF5F6C98A04030000000
00010FEFE3FDFAE6E903020C")::
 DISPLAY AT(6,8):"HAASISOFT
1986" :: DISPLAY AT(9,8):"
  ZEIGT:"
295 DISPLAY AT(1,1):" MICHAE
L HAASLER SOFTWARE" :: DISPL
AY AT(24,1):" MICHAEL HAASLE
R SOFTWARE"
300 FOR I=1 TO 1000 :: NEXT
I :: CALL CLEAR :: DISPLAY A
T(10,10):"K A I S E R" :: CA
LL SPRITE(#1,120,11,48,126):
: CALL MUSIC :: CALL CLEAR :
: CALL DELSPRITE(ALL)
330 DISPLAY AT(1,1):"......
K A I S E R" :: DISPLAY AT(3
,1):"STEUERN SIE IHR FÛERSTE
NTUM!" :: DISPLAY AT(5,1):"H
ANDELN SIE MIT LAND UND  "
360 DISPLAY AT(7,1):"KORN. S
ETZEN SIE DIE STEUERN" :: DI
SPLAY AT(9,1):"FEST. FUEHREN
 SIE KRIEG MIT" :: DISPLAY A
T(11,1):"ANDEREN STAATEN UND
 LENKEN"
390 DISPLAY AT(13,1):"SIE IH
R REICH SO, DASS SICH" :: DI
SPLAY AT(15,1):"DIE BEVOELKE
RUNG VERMEHRT "
410 DISPLAY AT(17,1):"UND SI
E BEFOERDERT WERDEN:"
420 DISPLAY AT(19,1):"BIS ZU
M >KAISER<" :: CALL MUSIC ::
 INPUT "WEITER MIT 1":WA ::
IF WA=1 THEN RUN "DSK1.KAISE
R2"
10000 SUB MUSIC :: RESTORE 1
0040
10011 READ A1,B1 :: IF B1=0
THEN SUBEXIT
10020 CALL SOUND(175*A1,B1,0
):: GOTO 10011
10040 DATA 3,349,1,262,3,392
,1,262,3,440,1,349,4,466,3,4
94,1,392,3,494,1,392,4,523
10050 DATA 1,587,1,659,1,698
,2,523,1,349,1,587,1,659,1,6
98,3,523
10060 DATA 1,587,1,659,1,698
,1,523,1,659,1,698,2,880,1,7
84,1,698,2,789,0,0
10070 SUBEND
```

len, den langsamen Aufbau des Palastes, die Größe des Landes und die Stärke des jeweiligen Heeres in einer Grafik darstellen, die auf Wunsch oder nach jedem Jahr automatisch abgerufen wird.
Wäre dies nicht ein Anreiz für die besten der TI-Programmierer, dieses Programm entsprechend umzuarbeiten, um den Spielspaß noch um ein Vielfaches zu erhöhen?
Auch TI-Clubs können sich in Gemeinschaftsarbeit gerne daran beteiligen, um Ihr Wissen und Können zu beweisen.
Fühlen Sie sich angesprochen, dann nichts wie ran an den Computer und umprogrammieren. Die Länge kann ruhig die Speichererweiterung ausfüllen, denn die meisten noch aktiven TI-User sind ohnehin gut ausgerüstet.
In unserem nächsten Sonderheft wird für die beste Version auf jeden Fall ein Ehrenplatz freigehalten und dem besten Programmierer oder Club winkt eine zusätzliche kleine Überraschung.

LISTINGS

```
10 ! **********************
11 ! *                    *
12 ! * STUFEN ZUM ERFOLG  *
13 ! *   Hauptprogramm    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *  Michael Haasler   *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *    11938 Bytes     *
28 ! *                    *
29 ! **********************
30 !
110 RANDOMIZE :: KAV,ART,INF
,KRE,SCH=0 :: MIL=5 :: RESTO
RE :: CALL CLEAR
120 DATA 003F666C746466EF,00
7E313B3633337E,003E634140603
F1E,007E321119113F7E,007F333
03C38317F,007F723239383078
130 DATA 003E6648504F663E,00
7C603F3132327A03,007C38181C1
A183C2,003F13060606263F
140 DATA 007B72363C363673,00
783030307263 7F
150 DATA 007F2B2A49494963,00
7C76323A323477,003E634559617
F3E,007E33313F3E3078,003E634
559617F3E07
160 DATA 007F313A34323179,00
3E413C06634F3E
170 DATA 007F492A0808081C,00
6F2626666E7E37,0073622232321
E0C,0077222241494936,0079321
40814264F
180 DATA 007341623418183C,00
7F43260C1A317F
200 FOR J=1 TO 160 :: NEXT J
 :: FOR J=65 TO 90 :: READ Z
$ :: CALL CHAR(J,Z$):: NEXT
J
241 CALL SCREEN(2):: FOR I=2
 TO 10 :: CALL COLOR(I,5,2):
: NEXT I :: Z$="HERR " :: J$
="NORMAL" :: ZO=3 :: ES=3 ::
 MS=3 :: M=0 :: F=0 :: P=0 :
: INPUT "NAME ?":NAME$ :: CA
LL CLEAR :: L=1000 :: V=1000
270 JAHR=1700 :: BEV=2000
280 ! ROUTINE
290 CALL CLEAR :: RANDOMIZE
:: CALL ZU(WETTER,1,6):: IF
WETTER=1 THEN WETTER$="TROCK
ENHEIT-SCHLECHTE ERNTE" :: G
OTO 300
291 IF WETTER=2 THEN WETTER$
="REGEN-MIESE ERNTE" :: GOTO
 301
292 IF WETTER=3 THEN WETTER$
="WETTER NORMAL-ERNTE NORMAL
" :: GOTO 302
293 IF WETTER=4 THEN WETTER$
="WETTER GUT-ERNTE GUT" :: G
OTO 303
294 IF WETTER=5 THEN WETTER$
="WETTER OPTIMAL-ERNTE OPTIM
AL" :: GOTO 304
300 CALL ZU(KQ,200,1000):: K
V=KV+KQ :: GOTO 310
301 CALL ZU(KQ,700,1500):: K
V=KV+KQ :: GOTO 310
302 CALL ZU(KQ,1000,2500)::
KV=KV+KQ :: GOTO 310
303 CALL ZU(KQ,2000,2800)::
KV=KV+KQ :: GOTO 310
304 CALL ZU(KQ,3000,4000)::
KV=KV+KQ :: GOTO 310
310 !
311 CALL ZU(KN,300,3000):: C
ALL ZU(KP,10,100):: CALL ZU(
LP,10,100)
313 DISPLAY AT(1,1):Z$;NAME$
;" VON LAUSITZ" :: DISPLAY A
T(3,1):"JAHR:";JAHR :: DISPL
AY AT(5,1):WETTER$ :: IF KV<
0 THEN KV=0
332 IF L<0 THEN L=0
333 IF V<-1000000 THEN 20000
340 DISPLAY AT(7,1):"KORNVOR
RAT:....";KV;"KG" :: DISPLAY
 AT(9,1):"KORNNACHFRAGE:.";K
N;"KG" :: DISPLAY AT(11,1):"
KORNPREIS:.....";KP;"TALER"
370 DISPLAY AT(13,1):"LANDPR
EIS:.....";LP;"TALER" :: DIS
PLAY AT(15,1):"LANDBESITZ:..
..";L;"HA" :: DISPLAY AT(17,
1):"VERMOEGEN:.....";V;"TALE
R"
400 DISPLAY AT(19,1):"DRUECK
E:1=KORN KAUFEN.............
..2=KORN VERKAUFEN..........
..3=LAND KAUFEN.............
..4=LAND VERKAUFEN..........
..5=WEITER"
410 INPUT "IHRE WAHL ?":WAHL
420 IF WAHL=1 THEN 430 ELSE
IF WAHL=2 THEN 440 ELSE IF W
AHL=3 THEN 450 ELSE IF WAHL=
4 THEN 460 ELSE 465
430 CALL CLEAR :: INPUT "WIE
VIEL KORN ?":A :: V=V-A*KP :
: KV=KV+A :: GOTO 313
440 CALL CLEAR :: INPUT "WIE
VIEL KORN ?":A :: V=V+A*KP :
: KV=KV-A :: GOTO 313
450 CALL CLEAR :: INPUT "WIE
VIEL LAND ?":B :: V=V-B*LP :
: L=L+B :: GOTO 313
460 CALL CLEAR :: INPUT "WIE
VIEL LAND ?":B :: V=V+B*LP :
: L=L-B :: GOTO 313
```

```
465 !
470 V=V+ZO*BEV/100 :: V=V+ES
*BEV/100 :: V=V+MS*BEV/100
471 IF J$="HARMLOS" THEN V=V
+BEV/1000 ELSE IF J$="NORMAL
" THEN V=V+BEV/100 ELSE IF J
$="STRENG" THEN V=V+BEV/10 E
LSE IF J$="GIERIG" THEN V=V+
BEV/5
480 CALL CLEAR :: DISPLAY AT
(1,1):Z$;NAME$;" VON LAUSITZ
" :: DISPLAY AT(5,1):"ZOLL:.
.........";ZO;" TALER" :: D
ISPLAY AT(7,1):"MEHRWERTSTEU
ER:.";MS;" TALER"
510 DISPLAY AT(9,1):"EINKOMM
ENSTEUER:";ES;" TALER" :: DI
SPLAY AT(11,1):"JUSTIZ:.....
....";J$
530 DISPLAY AT(13,1):"1=ZOLL
 AENDERN.............2=MEHR
WERTSTEUERN AENDERN...3=EINK
OMMENSTEUER AENDERN...4=JUST
IZ AENDERN...........5=WEIT
ER"
535 DISPLAY AT(19,1):"VERMOE
GEN";V :: INPUT "IHRE WAHL ?
":SW :: IF SW=1 THEN 560 ELS
E IF SW=2 THEN 570 ELSE IF S
W=3 THEN 580 ELSE IF SW=4 TH
EN 590 ELSE 600
560 CALL CLEAR :: INPUT "ZOL
L AENDERN:[1-1000]":ZO :: IF
 ZO>1000 THEN 560 ELSE 480
570 CALL CLEAR :: INPUT "M-W
 STEUERN AENDERN:[1-1000]":M
S :: IF MS>1000 THEN 570 ELS
E 480
580 CALL CLEAR :: INPUT "E-S
TEUERN AENDERN:[1-1000]":ES
:: IF ES>1000 THEN 580 ELSE
480
590 CALL CLEAR :: PRINT "JUS
TIZ AENDERN:[HARMLOS,....NOR
MAL,STRENG,GIERIG]" :: INPUT
 J$ :: GOTO 480
600 ! INVESTITIONEN
610 CALL CLEAR :: IF V<-1000
000 THEN 20000
620 DISPLAY AT(1,1):Z$;NAME$
;" VON LAUSITZ" :: DISPLAY A
T(3,1):"INVESTITIONEN:" :: D
ISPLAY AT(5,1):"1. MARKT:...
.....50000 TALER" :: DISPLAY
 AT(7,1):"2. FABRIK:.......6
0000 TALER"
660 DISPLAY AT(9,1):"3. PALA
ST:......100000 TALER" :: DI
SPLAY AT(11,1):"4. TRUPPEN:.
.....========>>>" :: DISPLAY
 AT(13,1):"5. KRIEG" :: DISP
LAY AT(15,1):"VERMOEGEN:";V
690 DISPLAY AT(17,1):"SIE HA
BEN:MAERKTE...:";M;"
  FABRIKEN..:";F;"
PALAESTE..:";P :: INPUT "IHR
E WAHL[6=WEITER]:":W
710 IF W=1 THEN 720 ELSE IF
W=2 THEN 730 ELSE IF W=3 THE
N 740 ELSE IF W=4 THEN 760 E
LSE IF W=5 THEN 761 ELSE 764
720 V=V-50000 :: M=M+1 :: GO
TO 600
730 V=V-60000 :: F=F+1 :: GO
TO 600
740 V=V-100000 :: P=P+1 :: G
OTO 600
760 GOSUB 1040 :: GOTO 600
761 GOSUB 1300 :: GOTO 600
762 !
763 !
764 CALL CLEAR :: IF L<(BEV*
4)THEN DISPLAY AT(12,1):"DIE
 BUERGER BRAUCHEN MEHR......
.........................LAN
D."
765 IF L<BEV*4 THEN BEV=BEV-
100
767 !
768 !
769 TS=(KAV*750)+(ART*600)+(
MIL*250)+(INF*500)+(KRE*250)
+(SCH*250):: DISPLAY AT(5,1)
:"DIE SOLDATEN ERHIELTEN IHR
EN.........................
..SOLD !"
771 DISPLAY AT(9,1):"ES WARE
N";TS;" TALER"
772 FOR I=1 TO 750 :: NEXT I
 :: V=V-TS :: IF KV<KN AND J
$="GIERIG" THEN 850
790 IF KV<KN AND J$="STRENG"
 THEN 850
800 IF KV<KN AND J$="NORMAL"
 OR J$="HARMLOS" THEN 860
810 IF KV>KN AND J$="GIERIG"
 THEN 870
820 IF KV>KN AND J$="STRENG"
 THEN 870
830 IF KV>KN AND J$="NORMAL"
 OR J$="HARMLOS" THEN 880
850 CALL ZU(GEB,100,200):: C
ALL ZU(STE,200,300):: CALL Z
U(EIN,100,200):: CALL ZU(AUS
,200,300):: GOTO 890
860 CALL ZU(GEB,100,500):: C
ALL ZU(STE,100,500):: CALL Z
U(EIN,100,300):: CALL ZU(AUS
,100,300):: GOTO 890
870 CALL ZU(GEB,200,300):: C
ALL ZU(STE,100,250):: CALL Z
U(EIN,200,400):: CALL ZU(AUS
,100,300):: GOTO 890
880 CALL ZU(GEB,200,300):: C
ALL ZU(STE,100,200):: CALL Z
U(EIN,200,300):: CALL ZU(AUS
```

```
,100,200):: GOTO 890
890 CALL CLEAR :: DISPLAY AT
(1,1):Z$;NAME$;" VON LAUSITZ
" :: DISPLAY AT(5,1):"GEBURT
EN:.......";GEB :: DISPLAY A
T(7,1):"STERBEFAELLE:...";ST
E :: DISPLAY AT(9,1):"EINWAN
DERER:....";EIN
930 DISPLAY AT(11,1):"AUSWAN
DERER:....";AUS :: BEV=BEV-A
US :: BEV=BEV-STE :: BEV=BEV
+EIN :: BEV=BEV+GEB :: IF BE
V<1 THEN BEV=1
940 DISPLAY AT(13,1):"BEVOEL
KERUNG:...";BEV :: FG=INT(RN
D*1500)+1 :: MG=INT(RND*1000
)+1 :: IF M>=1 THEN DISPLAY
AT(15,1):"DIE MAERKTE BRACHT
EN:";MG*M;" TALER EIN." :: V
=V+MG*M
960 IF F>=1 THEN DISPLAY AT(
18,1):"DIE FABRIKEN BRACHTEN
:";FG*F;"TALER EIN." :: V=V+
FG*F
965 JAHR=JAHR+1 :: IF BEV>30
00 AND BEV<4500 THEN Z$="BAR
ON " :: GOTO 1025
980 IF BEV>4501 AND BEV<6000
 THEN Z$="GRAF " :: GOTO 102
5
990 IF BEV>6000 AND BEV<7000
 THEN Z$="KURFUERST " :: GOT
O 1025
1000 IF BEV>7000 AND BEV<800
0 THEN Z$="HERZOG " :: GOTO
1025
1010 IF BEV>8000 AND BEV<100
00 THEN Z$="KOENIG " :: GOTO
 1025
1020 IF BEV>10000 THEN DISPL
AY AT(20,1):"SIE SIND ZUM >K
AISER<.......GEKROENT WORDEN
." :: Z$="KAISER "
1025 DISPLAY AT(22,1):"SIE S
IND DERZEIT EIN:";Z$
1030 INPUT "1=WEITER":A :: I
F A=1 THEN 2000 ELSE 1030
1040 CALL CLEAR :: DISPLAY A
T(1,1):"......AUFRUESTUNG" :
: DISPLAY AT(3,1):".....LAND
STREITKRAEFTE:" :: DISPLAY A
T(5,1):"1.KAVALLERIE:....500
0 TALER"
1080 DISPLAY AT(7,1):"2.ARTI
LLERIE:....4000 TALER" :: DI
SPLAY AT(9,1):"3.INFANTRIE:.
....3500 TALER" :: DISPLAY A
T(11,1):"4.MILIZ:.........20
00 TALER"
1110 DISPLAY AT(14,1):".....
SEESTREITKRAEFTE:" :: DISPLA
Y AT(16,1):"5.KREUZER:......
.50000 TALER" :: DISPLAY AT(
18,1):"6.SCHLACHTSCH....4000
0 TALER"
1135 DISPLAY AT(20,1):"VERMO
EGEN:";V;"TALER" :: INPUT "I
HRE WAHL ?[7=WEITER]":WAHL :
: IF WAHL=7 THEN 1200 ELSE 1
160
1160 IF WAHL=1 THEN V=V-5000
 :: KAV=KAV+1 :: GOTO 1040
1162 IF WAHL=2 THEN V=V-4000
 :: ART=ART+1 :: GOTO 1040
1164 IF WAHL=3 THEN V=V-3500
 :: INF=INF+1 :: GOTO 1040
1166 IF WAHL=4 THEN V=V-2000
 :: MIL=MIL+1 :: GOTO 1040
1168 IF WAHL=5 THEN V=V-5000
0 :: KRE=KRE+1 :: GOTO 1040
1170 IF WAHL=6 THEN V=V-4000
0 :: SCH=SCH+1 :: GOTO 1040
1200 CALL CLEAR :: DISPLAY A
T(1,1):"SIE HABEN:" :: DISPL
AY AT(3,1):"KAVALLERIE:";KAV
 :: DISPLAY AT(5,1):"ARTILLE
RIE:";ART :: DISPLAY AT(7,1)
:"INFANTRIE :";INF :: DISPLA
Y AT(9,1):"MILIZ.....:";MIL
1250 DISPLAY AT(11,1):"SCHLA
CTSCHIFFE:";SCH :: DISPLAY A
T(13,1):"KREUZER.......:";KR
E :: DISPLAY AT(15,1):"VERMO
EGEN:";V;"TALER" :: INPUT "W
EITER=1":WAHL :: IF WAHL=1 T
HEN RETURN
1300 ! KRIEG
1301 CALL CLEAR :: DISPLAY A
T(1,1):"GEGEN WEN WOLLEN SIE
.........KRIEG FUEHREN ????"
:: DISPLAY AT(5,1):"1.GROSS
BRITANIEN" :: DISPLAY AT(7,1
):"2.RUSSLAND" :: DISPLAY AT
(9,1):"3.FRANKREICH"
1350 DISPLAY AT(11,1):"4.DEU
TSCHLAND" :: DISPLAY AT(13,1
):"5.POLEN" :: INPUT "IHRE A
USWAHL ?":WAHL :: IF WAHL=1
THEN 1380 ELSE IF WAHL=2 THE
N 1400 ELSE IF WAHL=3 THEN 1
420 ELSE IF WAHL=5 THEN 1440
1371 IF WAHL=4 THEN 1460
1380 FMIL=100 :: FART=200 ::
 FKAV=200 :: FINF=150 :: FKR
E=121 :: FSCH=64 :: GOTO 160
0
1400 FMIL=150 :: FART=200 ::
 FKAV=150 :: FINF=400 :: FKR
E=14 :: FSCH=16 :: GOTO 1600
1420 FMIL=150 :: FART=250 ::
 FKAV=200 :: FINF=350 :: FKR
E=34 :: FSCH=28
1430 GOTO 1600
1440 FMIL=200 :: FART=75 ::
FKAV=100 :: FINF=100 :: FKRE
=6 :: FSCH=8 :: GOTO 1600
```

LISTINGS

```
146Ø CALL CLEAR :: DISPLAY A
T(1,1):"1.OESTERREICH" :: DI
SPLAY AT(3,1):"2.PREUSSEN" :
: DISPLAY AT(5,1):"3.SACHSEN
" :: DISPLAY AT(7,1):"4.BAYE
RN" :: DISPLAY AT(9,1):"5.HA
NNOVER"
1467 DISPLAY AT(11,1):"6.MEC
KLENBURG" :: DISPLAY AT(13,1
):"7.ANHALT" :: INPUT "IHRE
WAHL ?":WAHL :: IF WAHL=1 TH
EN 147Ø ELSE IF WAHL=2 THEN
148Ø ELSE IF WAHL=3 THEN 149
Ø ELSE IF WAHL=4 THEN 15ØØ
1469 IF WAHL=5 THEN 151Ø ELS
E IF WAHL=6 THEN 152Ø ELSE I
F WAHL=7 THEN 153Ø
147Ø FMIL=125 :: FART=225 ::
 FKAV=15Ø :: FINF=2ØØ :: FKR
E=12 :: FSCH=16 :: GOTO 16ØØ
148Ø FMIL=1ØØ :: FART=15Ø ::
 FKAV=125 :: FINF=175 :: FKR
E=42 :: FSCH=25 :: GOTO 16ØØ
149Ø FMIL=5Ø :: FART=75 :: F
KAV=1ØØ :: FINF=13Ø :: FKRE,
FSCH=Ø :: GOTO 16ØØ
15ØØ FMIL=45 :: FART=8Ø :: F
KAV=75 :: FINF=125 :: FKRE,F
SCH=Ø :: GOTO 16ØØ
151Ø FMIL=45 :: FART=85 :: F
KAV=85 :: FINF=1ØØ :: FKRE=4
 :: FSCH=2 :: GOTO 16ØØ
152Ø FMIL=4Ø :: FART=87 :: F
KAV=6Ø :: FINF=175 :: FKRE=5
 :: FSCH=1 :: GOTO 16ØØ
153Ø FMIL=2Ø :: FART=25 :: F
KAV=15 :: FINF=3Ø :: FKRE,FS
CH=Ø :: GOTO 16ØØ
16ØØ ! KRIEG
1605 CALL CLEAR :: DISPLAY A
T(1,1):".....K R I E G" :: D
ISPLAY AT(3,1):".........LA
USITZ <=> GEGNER" :: DISPLAY
 AT(5,1):"KAVALLERIE:";KAV;"
<=>";FKAV
164Ø DISPLAY AT(7,1):"ARTILL
ERIE:";ART;"<=>";FART :: DIS
PLAY AT(9,1):"INFANTRIE :";I
NF;"<=>";FINF :: DISPLAY AT(
11,1):"MILIZ.....:";MIL;"<=>
";FMIL
167Ø DISPLAY AT(13,1):"KREUZ
ER...:";KRE;"<=>";FKRE :: DI
SPLAY AT(15,1):"SCHLACHTSC:"
;SCH;"<=>";FSCH :: INPUT "1=
WEITER":WAHL :: IF WAHL=1 TH
EN 1685
1685 CALL ZU(ZUF,1ØØ,5ØØ)::
CALL ZU(FZUF,1ØØ,5ØØ)
169Ø GTR=(KAV*3)+(ART*2)+(IN
F*2)+MIL+(KRE*1Ø)+(SCH*6)+(Z
UF*5):: GFTR=(FKAV*3)+(FART*
2)+(FINF*2)+FMIL+(FKRE*1Ø)+(
FSCH*6)+(FZUF*5)
17ØØ IF GTR>GFTR THEN 18ØØ E
LSE 171Ø
171Ø ! KRIEG VERLOREN
1715 CALL CLEAR :: DISPLAY A
T(1,1):"SIE HABEN DEN KRIEG
VERLOREN" :: DISPLAY AT(3,1)
:"IHRE VERLUSTE:" :: CALL ZU
(ZBEV,1,1ØØØ):: BEV=BEV-ZBEV
1732 DISPLAY AT(5,1):"BEVOEL
KERUNG:";ZBEV;" TOTE" :: CAL
L ZU(ZV,1,5ØØØØØ):: V=V-ZV :
: DISPLAY AT(7,1):"VERMOEGEN
   :";ZV;" TALER" :: CALL ZU
(ZL,1,1ØØØØ):: L=L-ZL :: IF
L<Ø THEN L=Ø
174Ø DISPLAY AT(9,1):"LAND..
.....:";L;" HA" :: CALL ZU(
ZM,1,2Ø):: M=M-ZM :: IF M<Ø
THEN M=Ø
1744 DISPLAY AT(11,1):"MAERK
TE.....:";M :: CALL ZU(ZF,1,
2Ø):: F=F-ZF :: IF F<Ø THEN
F=Ø
1752 DISPLAY AT(13,1):"FABRI
KEN....:";F :: CALL ZU(ZMIL,
1,2Ø):: CALL ZU(ZART,1,2Ø)::
 CALL ZU(ZKAV,1,2Ø):: CALL Z
U(ZINF,1,2Ø):: CALL ZU(ZKRE,
1,2Ø):: CALL ZU(ZSCH,1,2Ø)
1761 MIL=MIL-ZMIL :: ART=ART
-ZART :: KAV=KAV-ZKAV :: INF
=INF-ZINF :: KRE=KRE-ZKRE ::
 SCH=SCH-ZSCH :: DISPLAY AT(
15,1):"KAVALLERIE :";ZKAV ::
 DISPLAY AT(17,1):"ARTILLERI
E :";ZART
1766 DISPLAY AT(19,1):"INFAN
TRIE  :";ZINF :: DISPLAY AT(
21,1):"MILIZ.....:";ZMIL ::
 INPUT "1=MEHR":MEHR :: IF M
EHR=1 THEN 1771
1771 CALL CLEAR :: DISPLAY A
T(1,1):"KREUZER.....:";ZK
RE :: DISPLAY AT(3,1):"SCHLA
CHTSCHIFFE:";ZSCH :: IF MIL<
Ø THEN MIL=Ø
1782 IF KAV<Ø THEN KAV=Ø
1783 IF ART<Ø THEN ART=Ø
1784 IF INF<Ø THEN INF=Ø
1785 IF KRE<Ø THEN KRE=Ø
1786 IF SCH<Ø THEN SCH=Ø
1799 INPUT "1=WEITER":WAHL :
: IF WAHL=1 THEN 6ØØ
18ØØ ! KRIEG GEWONNEN
1805 CALL CLEAR :: DISPLAY A
T(1,1):".....KRIEG GEWONNEN"
 :: DISPLAY AT(3,1):"SIE HAB
EN GEWONNEN:" :: CALL ZU(SBE
V,1,1ØØØ):: BEV=BEV+SBEV ::
DISPLAY AT(5,1):"BEVOELKERUN
G  :";SBEV
185Ø CALL ZU(SV,1,5ØØØØØ)::
```

```
V=V+SV :: DISPLAY AT(7,1):"V
ERMOEGEN.....:";SV;"TALER" :
: CALL ZU(SL,1,10000):: L=L+
SL :: DISPLAY AT(9,1):"LAND.
.........:";SL;"HA"
1890 CALL ZU(ZMIL,1,10):: CA
LL ZU(ZART,1,10):: CALL ZU(Z
KAV,1,10):: CALL ZU(ZINF,1,1
0):: CALL ZU(ZKRE,1,10):: CA
LL ZU(ZSCH,1,10)
1891 MIL=MIL-ZMIL :: ART=ART
-ZART :: KAV=KAV-ZKAV :: INF
=INF-ZINF :: KRE=KRE-ZKRE ::
 SCH=SCH-ZSCH :: DISPLAY AT(
11,1):"VERLUSTE:" :: DISPLAY
 AT(13,1):"KAVALLERIE :";ZKA
V
1896 DISPLAY AT(15,1):"ARTIL
LERIE :";ZART :: DISPLAY AT(
17,1):"INFANTRIE  :";ZINF ::
 DISPLAY AT(19,1):"MILIZ....
..:";ZMIL :: DISPLAY AT(21,1
):"KREUZER.........:";ZKRE
1920 DISPLAY AT(23,1):"SCHLA
CHTSCHIFFE:";ZSCH :: IF MIL<
0 THEN MIL=0
1922 IF KAV<0 THEN KAV=0
1923 IF INF<0 THEN INF=0
1924 IF ART<0 THEN ART=0
1925 IF KRE<0 THEN KRE=0
1926 IF SCH<0 THEN SCH=0
1950 INPUT "1=WEITER":WAHL :
: IF WAHL=1 THEN 600
2000 ! ANGRIFF ????
2010 CALL ZU(ANG,1,100):: IF
 ANG>50 AND ZO>500 AND MS>40
0 AND ES>300 THEN 2020 ELSE
280
2020 CALL CLEAR :: DISPLAY A
T(6,1):"DIE BAUERN SIND UNZU
FRIEDEN.....
.....ES GIBT EINEN AUFSTA
ND"
2030 CALL ZU(BAU,10,100):: D
ISPLAY AT(12,1):"IHRE MILIZ:
 <=> DIE BAUERN:" :: DISPLAY
 AT(14,1):" ";MIL;"
  ";BAU
2041 FOR I=1 TO 750 :: NEXT
I :: CALL ZU(AUFS,100,500)::
 CALL ZU(AUFT,100,500):: MIL
S=MIL+AUFS :: BAUS=BAU+AUFT
:: IF BAUS>MILS THEN 2100 EL
SE 2070
2070 CALL CLEAR :: DISPLAY A
T(1,1):"AUFSTAND NIEDERGESCH
LAGEN" :: BEV=BEV-BAU :: DIS
PLAY AT(3,1):"ES GAB";BAU;"T
OTE."
2095 FOR I=1 TO 750 :: NEXT
I :: GOTO 280
2100 CALL CLEAR :: DISPLAY A
T(1,1):"AUFSTAND ERFOLGREICH
 !!!" :: ZO,MS,ES=0 :: V=V-2
500000 :: DISPLAY AT(3,1):"25
0000 TALER ABTRITT AN DIE...
.........................BA
UERN !!!"
2120 MIL=MIL-5 :: FOR I=1 TO
 750 :: NEXT I :: GOTO 280
20000 ! BANKROTT
20010 CALL CLEAR :: DISPLAY
AT(8,4):"SIE SIND BANKROTT!!
!" :: V=V+M*25000 :: M=0 ::
V=V+F*300000 :: F=0 :: V=V+P*
50000 :: P=0 :: V=V+K*600000
:: K=0 :: V=V+T*25000 :: T=0
20030 IF V<=-10000000 THEN EN
D
20040 DISPLAY AT(10,1):"IHR
BESITZ WIRD GEPFAENDET !" ::
 FOR I=1 TO 1000 :: NEXT I :
: GOTO 280
30100 SUB ZU(A,B,C):: A=INT(
(C-B+1)*RND)+B :: SUBEND
```

BÖRSE

Verk. TI 99/4A m. viel Zubehör z.B. Speichererw. Ex. Basic 2 Plus. Tel: 0211/422216

Verk. TI 99/4A + Pac Man + Kleingrafiken + 92 Spiele + TI-Fachzeitschr. + Rec.-Kabel + Infocass. + Handbuch + Anleitung f. VB 250,– DM. Tel: 08142/52873, ab 15.00 Uhr

---

TI 99/4A: Ext. Doppellaufw. i. Box + ext. Disc. (Texas) + Diskmanager 680,–; RS 232 (Atronic) 2 Ports ext. 180,–; 32 K + Centronic (Atronic) ext. 160,–; Tips + Tricks 20,–; Spielen, lernen, arbeiten 20,–. Chipprogr. 7,– 77 Basicprogr. Lon Poole 15,– DM. Tel: 06182/26186, Harald Ruppert

---

Suche das Adventuremodul. Erwin Kinslechner, Eibesbrunnergasse 1/10, A-1120 Wien

---

Verk. Pers. Fin, Aids + Bas.-Progr. Rout + Market simulation + Oldies II + Hardcopyroutinen, je 20,– DM. Tel: 08708/759, Semm

---

Verk. Mini Mem., Editor Ass. Mulitplan, je 150,– DM, Tex-Forth 60,– DM. Tel: 08708/759, Semm

---

Verk. TI 99/4A + Ext. Basic + dt. Handbuch + Tips + Tricks + Rec.-Kabel 250,– DM. Tel: 05371/16150, ab 18.00 Uhr

---

Editor Ass. Buch 30,–; Modulator, Ch 36 + Fbas 40,–; Tastatur 20,–. 5 x Spiel Modulen, Parsec usw. 150,– DM. Tel: 06155/4692

!Zu Verschenken! habe ich nichts. Aber fast: TI 99/4A kompl. System. Liste geg. Freiumschl. J. Müller, Espanstr. 84, 8510 Fürth

---

Hallo TI-User! Wenn Ihr Euch einem Userclub anschließen wollt, dann denkt an REX-SOFT! Wir versorgen Euch u.a. m, Ass.-Freeware. Eine Info gibt es gegen Rückporto bei: Rex-Soft, Pestalozzistr. 7, 2090 Winsen/Luhe

---

Verk. Universaldrucker Cosmos JP 80. Voll grafikfähig, Aufl. 640/1280 Punkte/Zeile und voll Epson kompatibel, 9 Nadeln vertikal, 30 Mio. Zeichen, VB 500,– DM. Tel: 0202/783901, ab 18 Uhr

---

Verk. Orig. TI-Diskettanlaufw. 90 K-RAM f. TI-Box. 360,– DM. Tel: 0221/84 3625

TI 99/4A 150,–, dazu Ex-Basic, nicht einzeln! 150,– DM, TI-Intern 25,– Apple kompl. System 2000,– DM. Tel: 089/ 60 93425

Verk. TI 99/4A Anlage. Voll ausgeb. Einzeln o. zusammen. Info: Tel: 02331/586672

HALLOWEEN X

Dieses Spiel lehnt sich von der Idee her an den Film Halloween an. In diesem Film trieb ein menschliches Wesen in Verkleidung, wie sie zu Halloween üblich ist, sein Unwesen.
Trotz verschärfter Schutzmaßnahmen und Angriffsstrategien war es nicht möglich, dieses Monster zu besiegen.
Es schien so, als wenn dieses Wesen mehrere Leben hätte.
Es tankte immer wieder, auf unerklärliche Weise, Lebensenergie.
Das Ziel dieses Spiels ist es nun, die festgelegte Punktzahl von 100 Punkten zu erreichen, die die hundert Leben des Monsters wiederspiegeln.
Erreicht man weniger Punkte, kommt eine entsprechende Meldung, die einem eine Bewertung des Spielstandes anzeigt.
Um die Widerspenstigkeit und Heimtücke von Halloween X zu verdeutlichen wird darauf hingewiesen, daß dieses eigenartige Wesen sich auch unsichtbar nähern kann.

HASENJAGD

Hasenjagd ist ein in TI-Basic geschriebenes Simulationsprogramm. Es simuliert das Jäger-Beute-Gleichgewicht in der Natur. Dabei werden echte Wahrscheinlichkeiten für die Zahl der Nachkommen dargestellt. Außerdem wird auch der begrenzte Lebensraum für Hasen und Füchse mitberücksichtigt. Die Füchse, welche die Hasen jagen, haben unterschiedlich viele Fangversuche. Die Anzahl der Hasen und Füchse sowie die Zahl der Fangversuche werden zu Beginn eingegeben.
Es lohnt sich, mit diesen Ausgangsbedingungen zu experimentieren. Es können maximal 12 Generationen (Versuche) beobachtet werden.
Die Anzahl der Versuche ist frei wählbar. Am Ende jeder Generation werden die neuen Werte für die nächste Generation ausgegeben.
Zuerst (am Beginn der Generation) streiten sich die Hasen um den begrenzten Lebensraum. Derjenige, welcher keinen Raum besetzen kann, ist von Beginn an ausgeschieden, d.h., er kann sich nicht fortpflanzen. Hasen, die vom Fuchs geschlagen wurden, werden durch das Fuchssymbol ersetzt.
Es können sich nur die überlebenden Hasen (mit Faktor 5) und die Füchse (Faktor 2) fortpflanzen, welche einen Hasen erbeutet haben.
Die Auswertung der Simulation erfolgt am Ende mit Hilfe eines Balkendiagrammes. Es zeigt den Verlauf der beiden Populationen und den Verlauf der Vermehrungsraten. Beide sollen phasenverschobene Sinusfunktionen sein.
Interessante Versuchsergebnisse lassen sich auf Band speichern und auch wiedergeben.

LISTINGS

```
100 ! ************************
110 ! *                      *
120 ! *    HALLOWEEN X       *
130 ! *                      *
140 ! *   COPYRIGHT BY       *
150 ! *                      *
160 ! *   RALF FELDHOFF      *
170 ! *                      *
180 ! * BENOETIGTE GERAETE   *
190 ! *   TI99/4A KONSOLE    *
200 ! *    EXT. BASIC        *
210 ! *    JOYSTICK 1        *
220 ! *                      *
230 ! *  SPEICHERBELEGUNG    *
240 ! *     5838 BYTES       *
250 ! *                      *
260 ! ************************
270 CALL CLEAR
280 CALL CHAR(96,"31425673FF
EACB873651FEA51BC261FE267541
ACF14526661B647EFDAB846735")
290 CALL CHAR(140,"FFFFFFE3E
FFBF1FBFEFEFFE7E1F8FFFFFFFF
FC7F7DF8FDF7F7FFFE7871FFFFF"
)
300 CALL CHAR(110,"112233447
7880000")
310 CALL CHAR(92,"1232435467
5647854636FFD25663FC53E35635
42553542566615232DCF245376")
320 CALL CHAR(136,"0103070F0
3030303030307 0F1F3FFFFF80C0E
0F0C0C0C0C0C0C0E0F0F8FCFFFF"
)
330 CALL CHAR(128,"0000000000
103040E0F0403010000000000000
0008040201 0F020408000000000"
)
340 BL$=" B L I N D F I S C
H !"
350 AN$=" A N F A E N G E R
!"
360 AM$=" A M A T E U R !"
370 IN$=" I N S I D E R !"
380 PR$=" P R O F I  !"
390 REM WERBUNG
400 FOR X=1 TO 16
410 CALL SPRITE(#X,140,X,10*
X,12*X)
420 CALL MAGNIFY(4)
430 CALL SOUND(10,440,0)
440 NEXT X
450 FOR X=1 TO 400 :: NEXT X
460 CALL DELSPRITE(ALL)
470 FOR A=1 TO 9
480 CALL COLOR(9,2,A+3)
490 CALL HCHAR(10-A,5,96,A)
500 CALL HCHAR(10+A,5,96,A)
510 CALL HCHAR(10-A,27-A,96,
A)
520 CALL HCHAR(10+A,27-A,96,
A)
530 NEXT A
540 CALL HCHAR(10,13,96,1)
550 CALL HCHAR(10,18,96,1)
560 CALL HCHAR(13,15,96,2)
570 CALL HCHAR(14,12,96,1)
580 CALL HCHAR(14,19,96,1)
590 CALL HCHAR(15,13,96,1)
600 CALL HCHAR(16,14,96,1)
610 CALL HCHAR(17,15,96,1)
620 CALL HCHAR(17,16,96,1)
630 CALL HCHAR(16,17,96,1)
640 CALL HCHAR(15,18,96,1)
650 FOR B=1 TO 500 :: NEXT B
660 DISPLAY AT(10,8):"HALLOW
EEN X"
670 FOR C=1 TO 2
680 FOR D=4 TO 16
690 CALL SCREEN(3)
700 CALL COLOR(9,D,D-2)
710 CALL COLOR(14,20-D,D-1)
720 CALL HCHAR(9,8,140,15)
730 CALL HCHAR(11,8,140,15)
740 CALL VCHAR(9,8,140,3)
750 CALL VCHAR(9,23,140,3)
760 CALL SOUND(50,D*10+110,0
,550,0,660+D*10,0)
770 NEXT D
780 NEXT C
790 FOR V=1 TO 400 :: NEXT V
800 CALL CLEAR
810 PRINT "COPYRIGHT BY RALF
 FELDHOFF",,,:"HUETTENSTR. 3
7",,,:"4000 DUESSELDORF 1",,
,:"TEL. 0211 / 37 77 95"
820 FOR E=1 TO 1000 :: NEXT
E
830 CALL CLEAR
840 PRINT "ES EXISTIEREN 4 H
ARMLOSE" :: PRINT "HALLOWEEN
 UND HALLOWEEN X." :: PRINT
,,"DEINE AUFGABE:SUCHE UND"
850 PRINT "ZERSTOERE HALLOWE
EN X !" :: PRINT ,,"DU BEKOM
MST EINEN PUNKT" :: PRINT "(
LINKS UNTEN) WENN DU IHN" ::
 PRINT "ZERSTOERST."
860 PRINT "DU BEKOMMST EINEN
 " :: PRINT "FEHLERPUNKT(REC
HTS OBEN)" :: PRINT "WENN DU
 IHN NICHT TRIFFST" :: PRINT
 "ODER ER DICH ZERSTOERT."
870 PRINT ,,"DU HAST 5 VERSU
CHE." :: PRINT "DAS SPIEL EN
DET WENN DU ES" :: PRINT "IN
NERHALB DER 5 VERSUCHE" :: P
RINT "NICHT SCHAFFST HALLOWE
EN X"
880 PRINT "ZU ZERSTOEREN." :
: PRINT "ACHTUNG!!! :HALLOWE
EN X KANNSICH UNSICHTBAR NAE
HERN." :: PRINT ,,"DRUECKE D
EN FEUERKNOPF !!!"
890 CALL KEY(1,K,S)
900 IF K<>18 THEN 890
```

LISTINGS

```
910 REM SPIELFELDAUFBAU
920 CALL CLEAR
930 DISPLAY AT(4,10):"HALLOW
EEN X"
940 DISPLAY AT(22,5):"POINTS
:"
950 CALL COLOR(4,16,2)
960 CALL HCHAR(3,3,58,29)
970 CALL HCHAR(24,3,58,29)
980 CALL VCHAR(3,3,58,21)
990 CALL VCHAR(3,31,58,21)
1000 CALL COLOR(10,2,9)
1010 CALL HCHAR(1,2,110,31)
1020 CALL HCHAR(2,2,110,30)
1030 CALL VCHAR(2,32,110,22)
1040 CALL VCHAR(2,2,110,22)
1050 CALL MAGNIFY(3)
1060 CALL SPRITE(#17,136,2,1
47,147)
1070 RANDOMIZE
1080 FOR F=0 TO 99
1090 DISPLAY AT(22,12):F
1100 FOR R=19 TO 23
1110 DISPLAY AT(10,10):"HALL
OWEEN X" :: DISPLAY AT(12,10
):"TANKT ENERGIE !"
1120 CALL SPRITE(#R,140,RND*
15+1,RND*191+1,RND*255+1,RND
*20+F,RND*15+F)
1130 CALL JOYST(1,X1,Y1)
1140 CALL MOTION(#17,-5*Y1,5
*X1)
1150 NEXT R
1160 FOR G=0 TO 4
1170 DISPLAY AT(4,24):G
1180 DISPLAY AT(10,10):"....
......." :: DISPLAY AT(12,10
):"                "
1190 CALL JOYST(1,X1,Y1)
1200 CALL KEY(1,K,S)
1210 IF K=18 THEN 1250
1220 CALL MOTION(#17,-5*Y1,5
*X1)
1230 CALL COINC(#17,#19,15,Y
)
1240 IF Y=-1 THEN 1550 ELSE
1190
1250 CALL POSITION(#17,X,Y):
: CALL SPRITE(#18,130,2,INT(
ABS(X-4)+1),INT(ABS(Y-1)+1))
1260 CALL MOTION(#18,-127,0)
1270 FOR J=1 TO 4
1280 CALL COINC(#18,#19,15,Y
)
1290 IF Y=-1 THEN 1330
1300 NEXT J
1310 CALL DELSPRITE(#18)
1320 GOTO 1460
1330 CALL DELSPRITE(#18):: C
ALL SOUND(300,-6,0):: CALL P
OSITION(#19,X,Y):: CALL DELS
PRITE(#19):: CALL SPRITE(#20
,92,2,X,Y,0,0)
1340 FOR K=1 TO 100 :: NEXT
K :: CALL DELSPRITE(#20)
1350 NEXT F
1360 CALL DELSPRITE(#17)
1370 DISPLAY AT(22,12):F
1380 FOR S=110 TO 850 STEP 1
10
1390 CALL SOUND(-S,S,2,S+100
,2,S+200,2)
1400 NEXT S
1410 PRINT " **** H A L L O
W E E N ****" :: PRINT :: PR
INT "......M E I S T E R !"
1420 PRINT
1430 PRINT "DU HAST MIT 100
PUNKTEN",,,"DAS ZIEL ERREICH
T." :: PRINT ,"WILLST DU NOC
HMAL SPIELEN ?"
1440 PRINT ,"DANN DRUECKE DE
N FEUERKNOPF." :: PRINT ,,,,
"VIEL ERFOLG !!!"
1450 GOTO 1530
1460 IF G<>4 THEN 1620
1470 CALL DELSPRITE(#17)
1480 REM ENDMELDUNGEN
1490 CALL SOUND(100,110,0,22
0,0,330,0):: GOSUB 1640
1500 PRINT ,"DER LETZTE FEHL
ER WAR EIN" :: PRINT "FEHLSC
HUSS."
1510 PRINT ,"HALLOWEEN X HAT
 DICH BESIEGT",,,"WILLST DU
WEITERSPIELEN ?"
1520 PRINT ,"DANN DRUECKE DE
N FEUERKNOPF." :: PRINT ,,,,
"VIEL ERFOLG !!!"
1530 CALL KEY(1,CODE,STATUS)
1540 IF CODE=18 THEN 920 ELS
E 1530
1550 CALL SOUND(300,-6,0)::
CALL POSITION(#17,X,Y):: CAL
L DELSPRITE(#17):: CALL SPRI
TE(#20,92,2,X,Y,0,0)
1560 FOR L=1 TO 100 :: NEXT
L :: CALL DELSPRITE(#20):: C
ALL SPRITE(#17,136,2,147,147
,0,0):: GOTO 1610
1570 CALL DELSPRITE(#17):: G
OSUB 1640
1580 PRINT ,"DER LETZTE FEHL
ER WAR DER " :: PRINT "ZUSAM
MENSTOSS MIT" :: PRINT "HALL
OWEEN X."
1590 PRINT ,"HALLOWEEN X HAT
 DICH BESIEGT",
1600 PRINT ,"WILLST DU WEITE
RSPIELEN ?" :: PRINT ,"DANN
DRUECKE DEN FEUERKNOPF." ::
PRINT ,,,,"VIEL ERFOLG !!!"
:: GOTO 1530
1610 IF G=4 THEN 1570
1620 NEXT G
1630 REM LEISTUNG
```

```
1640 IF F=0 THEN PRINT BL$ :
: GOTO 1690
1650 IF F<20 THEN PRINT AN$
:: GOTO 1690
1660 IF F<40 THEN PRINT AM$
:: GOTO 1690
1670 IF F<60 THEN PRINT IN$
:: GOTO 1690
1680 PRINT PR$
1690 PRINT
1700 RETURN
```

LISTINGS

BÖRSE

Neuw. Software zu verk.! Module: Dig Dug 15,–, Return to Pirate Isle 25,–, Parsec 20,–, Munch Man 20,–, TI Invaders 20,–, Chisholm Trail 10,–, Alpiner 20,–, Adv. + allen 11 Cass. 75,–, 70 Cass. m. ca. 400 Progr., tägl. 15 - 17 Uhr Volker Niemeyer, Tel: 04202/81279

Suche TI-User zwecks Erfahrungsaustausch im Raum Düren. Stefan Bachem, Arnoldsweilerstr. 52, 5160 Düren, Tel: 02421/17673

Schnellste Dateiverw. m. Kass. Ex-Basic-Progr. auf Kass. geg. Einsendung von 10,– DM. Peter Hielscher, Am Wall 22, 4401 Saerbeck

---

Verk.:
Konsole TI 99/4A, s/w Monitor, Speech Synthesizer, TI-Box m. Flex Cable Karte, 32 KB Karte, RS232 Schnittstellen Karte (1xParallel, 2 x seriell), Disk-Controller, 5 1/4 Disklaufw. (alles orig. TI)
Modulsoftware: Diskmanager II, Ex. Basic, Ed.-Ass., Multiplikation I, Add. u. Subtr., Minus Miss., Parsec, Alpiner (alles orig. TI)
Literatur: 99 Spezial I, Ed. Assembl. Handbuch (dt. u. engl.), Mini Memory Handbuch, TI-Rev. (15 Hefte), Computer Contact (7 Hefte)
Ca. 10 Disk. m. Software, Kass.Rec.Kabel (doppelt), Joystick, Druckerkabel (parallel), Schaltpläne. Preis– VB. Neuw. ca. 3500,– DM, Tel: 06403/2289, ab 16 Uhr

---

Verk. 80-Zeichen-Drucker Ti-Silent 743 m. RS 232-Anschlußkabel (orig. TI) u. Papier 250,–. Original TI-RS232 Schnittstelle (ext.), 32K-Erw. eingebaut! 500,– DM. M. Becker, Tel: 06201/16272

2K-Speicher für nur 100,– direkt ansteckbar zu best. bei: C. Mohr, Friesenweg 38, 4133 Neuk.-Vluyn, Tel: 02845/21507

32K-Erw. z. Einbau i.d. Konsole (Superschnell, da direkt. Anschluß aller 16 Datenleit.) Lötarbeit erforderl. Verk. auch einige Module sowie Buch 99 Special. Preis: VB! Tel: 06257/83247

Wenn die 256K-Speichererweiterung lieferbar ist, erscheint im Clubheft des TI-CLUB BAUNATAL der ausführlichste Testbericht dieser Erweiterung. Wollen Sie den Bericht lesen? Ja, dann fordern Sie noch heute geg. 3,– DM Clubhefte an – bei: TI-CLUB BAUNATAL, M. Orf, Birkenallee 34, D-3407 Baunatal 1, Tel: 0561/49 7990

Orig. TI-Erweiterungsbox, 32K, Disc-Kontroller, 1 Int. und ein ext. Laufwerk, RS 232-Karte, Buchhaltungsmodul, Personal Rec. Keeping + Maker, Schaltunterlagen, versch. Literatur, sehr viel Software auf Disk., Mathematik 1, Basic Progr. Routinen II u. III, Baustatistik, kompl. abzugeb. für 1400,– DM. Günter Trunk, Urnenstr. 9, 6700 Ludwigshafen 25, Tel: 0621/678578, ab 17.00 Uhr

---

Endlich ist es da!!! Dt. Handbuch für TI-Writer u. TI-Forth. Preis pro Buch: 45,– DM. Ralph Weber, Langemarckstr. 54, 8906 Gersthofen

---

Achtung! Tausche Mini Mem. orig. TI + Literatur geg. Sprachsynthes. M. Macke, Tel: 0431/526083

Verk. TI 99/4A f. 100,– DM sowie versch. TI-Rev.-Hefte zum halben Preis. Tel: 098 71/9835

```
10 REM*********************
11 REM*    HASENJAGD      *
12 REM*Computersimulation*
13 REM*  Copyright  by    *
14 REM*                   *
15 REM*   Hans Enderl     *
16 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM* opt.:Kas.-Rekord.*
20 REM*                   *
23 REM* Speicherbelegung *
24 REM*    4125 Bytes     *
25 REM*********************
26 REM
100 CALL CLEAR
110 PRINT TAB(8);"MENUE":""
120 PRINT "1=>NEUE VERSUCHE"
:"":"2=>DATENSPEICHERUNG":""
:"3=>DATEN VOM BAND":" "
130 INPUT "KENNZIFFER ? :":C
140 IF C>3 THEN 130
150 IF C<1 THEN 130
160 ON C GOSUB 170,1780,1660
170 CALL CLEAR
180 PRINT TAB(10);"HASENJAGD
": :"SIMULATION DER POPULATI
ONEN": :"WAEHLEN SIE :":"1=>
ANZAHL DER VERSUCHE":"MAXIMA
L  12":""
190 PRINT "2=>ANFANGSZAHL DE
R HASEN": :"3=>ANFANSZAHL DE
R FUECHSE": :"4=>FANGVERSUCH
E PRO FUCHS":"MAX.4 0=ENDE":
""
200 PRINT "DER LEBENSRAUM IS
T 621 PLAETZE  GROSS.":""
210 PRINT "BITTE TASTE DRUEC
KEN !"
220 CALL KEY(0,KEY,STAT)
230 IF STAT=0 THEN 220
240 INPUT "1=>ANZAHL DER VER
SUCHE ?":V
250 IF V>12 THEN 240
260 IF V<=0 THEN 1650
270 INPUT "2=>ANFANGSZAHL HA
SEN ?":H
280 INPUT "3=>ANFANGSZAHL FU
ECHSE ?":F
290 INPUT "4=>FANGVERSUCHE P
RO FUCHS ?":VE
300 IF VE>4 THEN 290
310 PRINT "EINGABE IN ORDNUN
G ? (J/N)"
320 CALL KEY(0,KEY,STAT)
330 IF KEY=74 THEN 360
340 IF KEY=78 THEN 240
350 GOTO 320
360 DIM H1(12,2),F1(12,2)
370 CALL CLEAR
380 CALL COLOR(9,2,7)
390 CALL COLOR(2,2,3)
400 CALL SCREEN(13)
```

LISTINGS

```
410 CALL CHAR(100,"0123C3227
C7C2482")
420 CALL CHAR(42,"24243C183C
7E7E3C")
430 FOR V=1 TO V
440 K$="VERSUCH : "&STR$(V)
450 FOR U=1 TO LEN(K$)
460 CALL HCHAR(24,U+9,ASC(SE
G$(K$,U,1)))
470 NEXT U
480 V=VAL(STR$(V))
490 RANDOMIZE
500 FOR N=1 TO H
510 A=INT((23-1+1)*RND)+1
520 B=INT((29-3+1)*RND)+3
530 CALL GCHAR(A,B,WW)
540 IF WW=42 THEN 570
550 CALL HCHAR(A,B,42,1)
560 HH=HH+1
570 NEXT N
580 FOR N=1 TO F*VE
590 X=INT((23-1+1)*RND)+1
600 Y=INT((29-3+1)*RND)+3
610 CALL GCHAR(X,Y,WW)
620 IF WW=42 THEN 630 ELSE 6
50
630 CALL HCHAR(X,Y,100,1)
640 G=G+1
650 NEXT N
660 CALL CLEAR
670 H=(HH-G)*5
680 F=G*2
690 FOR Y=1 TO 3
700 FOR X=1500 TO 2500 STEP
250
710 CALL SOUND(-1000,X,0,-1,
0)
720 NEXT X
730 NEXT Y
740 PRINT TAB(10);"ERGEBNIS"
:"":"HASEN :";HH,"GEFRESSEN:
";G:""
750 PRINT
760 PRINT "IN DER NAECHSTEN
GENERATION"
770 PRINT
780 PRINT "HASEN :";H,"FUECH
SE :";F
790 PRINT
800 PRINT
810 PRINT "UNTERBRECHUNG TAS
TE DRUECKEN"
820 H1(V,1)=INT(HH/27)
830 H1(V,2)=INT(H/100)
840 F1(V,1)=INT(G/27)
850 F1(V,2)=INT(F/100)
860 FOR T=1 TO 500
870 CALL KEY(0,KEY,STAT)
880 IF STAT=0 THEN 890
885 GOTO 960
890 NEXT T
900 CALL CLEAR
910 G=0
920 HH=0
930 IF F<=0 THEN 960
940 IF H<=0 THEN 960
950 NEXT V
960 REM BALKENGRAFIK
970 CALL CLEAR
980 CALL SCREEN(10)
990 R$="0103070F1F3F7FFF"
1000 Q$="FFFFFFFFFFFFFFFF"
1010 CALL CHAR(42,"24243C183
C7E7E3C")
1020 CALL CHAR(100,"0123C322
7C7C2482")
1030 CALL CHAR(144,Q$)
1040 CALL CHAR(152,R$)
1050 CALL CHAR(104,Q$)
1060 CALL CHAR(112,Q$)
1070 CALL CHAR(120,R$)
1080 CALL CHAR(128,R$)
1090 CALL CHAR(136,R$)
1100 CALL COLOR(10,6,2)
1110 CALL COLOR(11,12,2)
1120 CALL COLOR(12,10,12)
1130 CALL COLOR(13,2,1)
1140 CALL COLOR(14,12,2)
1150 CALL COLOR(15,9,2)
1160 CALL COLOR(16,2,6)
1170 CALL COLOR(2,2,1)
1180 CALL COLOR(9,2,1)
1200 FOR I=621 TO 1 STEP -27
1210 PRINT I
1220 NEXT I
1225 FOR N=1 TO 2
1230 IF N=1 THEN 1240 ELSE 1
260
1240 K$="POPULATIONEN"
1250 GOTO 1270
1260 K$="VERMEHRUNGEN"
1270 FOR U=1 TO LEN(K$)
1280 CALL HCHAR(1,U+9,ASC(SE
G$(K$,U,1)))
1290 NEXT U
1300 P=7
1310 FOR V=1 TO V-1
1320 IF H1(V,N)>23 THEN 1330
 ELSE 1350
1330 Z=23
1340 GOTO 1380
1350 Z=H1(V,N)
1360 IF Z<1 THEN 1370 ELSE 1
380
1370 Z=1
1380 CALL VCHAR(24-Z,P,104,Z
)
1390 CALL VCHAR(24-Z,P+1,112
,Z-1)
1400 CALL HCHAR(23,P+1,120)
1410 CALL HCHAR(24-Z+1,P,42,
1)
1420 CALL HCHAR(24-Z,P+1,136
)
1430 CALL HCHAR(24-Z,P,128)
1440 IF F1(V,N)>24 THEN 1450
```

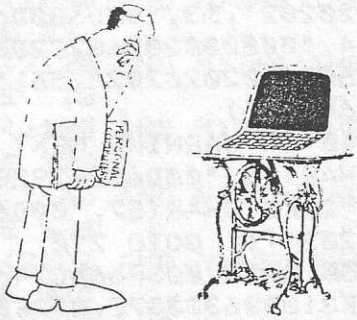
```
 .ELSE 1470
1450 Z=23
1460 GOTO 1500
1470 Z=F1(V,N)
1480 IF Z<1 THEN 1490 ELSE 1
500
1490 Z=1
1500 CALL VCHAR(24-Z,P,144,Z
)
1510 CALL VCHAR(24-Z,P+1,112
,Z-1)
1520 CALL HCHAR(23,P+1,120)
1530 CALL HCHAR(24-Z+1,P,100
,1)
1540 CALL HCHAR(24-Z,P+1,136
)
1550 CALL HCHAR(24-Z,P,152)
1560 P=P+2
1570 NEXT V
1580 CALL KEY(0,KEY,STAT)
1590 IF STAT=0 THEN 1580
1600 FOR I=1 TO 24
1610 CALL HCHAR(I,7,32,27)
1620 NEXT I
1630 NEXT N
1640 GOTO 100
1650 END
1660 REM DATENEINGABE
1670 PRINT TAB(7);"DATENEIGA
BE VOM BAND":""
1680 OPEN #1:"CS1",SEQUENTIA
L,INTERNAL,INPUT ,FIXED
1690 INPUT #1:V
1700 FOR N=1 TO 2
1710 FOR V=1 TO V-1
1720 INPUT #1:H1(V,N),F1(V,N
)
1730 NEXT V
1740 NEXT N
1750 CLOSE #1
1760 CALL CLEAR
1770 GOTO 960
1780 REM DATENSPEICHERUNG
1790 CALL CLEAR
1800 PRINT TAB(7);"DATENSPEI
CHERUNG"
1810 PRINT
1820 OPEN #1:"CS1",SEQUENTIA
L,INTERNAL,OUTPUT,FIXED
1830 PRINT #1:V
1840 FOR N=1 TO 2
1850 FOR V=1 TO V-1
1860 PRINT #1:H1(V,N),F1(V,N
)
1870 NEXT V
1880 NEXT N
1890 CLOSE #1
1900 GOTO 100
```

COMPUTER-TITEL AUS DEM VERLAG

LISTINGS

```
10 ! **********************
11 ! *                    *
12 ! *   Mendels Gesetz   *
13 ! * Computersimulation *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *    Hans Enderl     *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     4172 Bytes     *
28 ! *                    *
29 ! **********************
30 !
100 CALL CLEAR :: ON WARNING
 NEXT :: CALL INIT :: CALL L
OAD(-31806,16)
110 DISPLAY AT(11,1):"   MEN
DELSCHE REGELN"
120 DISPLAY AT(12,1):"   ===
==============="
130 CALL KEY(0,K,S):: IF S=0
 THEN 130
140 CALL CLEAR
142 DISPLAY AT(1,1):"IN DER
SIMULATION WIRD DER":"ERBGAN
G IN DER F2 GENERATION":"SIM
ULIERT.ES WERDEN JEWEILS":"4
00 KREUZUNGEN VON TIEREN":"D
ER F1 VORGENOMMEN."
144 DISPLAY AT(6,1):"BEI BSP
. 1":"R = ROT DOMINANT":"W =
 WEISS REZESSIV"
146 DISPLAY AT(9,1):"BEI BSP
. 2":"S = SCHWARZ DOMINANT":
"E = EINFARBIG REZESSIV":"r
= ROT REZESSIV":"s = GESCHEC
KT REZESSIV"
150 DISPLAY AT(16,1):"1 =>DO
MINATER ERBGANG DER F2  (MIT
 EINEM MERKMAL)"
160 DISPLAY AT(19,1):"2 =>DO
MINATER ERBGANG DER F2  (MIT
 ZWEI MERKMALEN)"
170 DISPLAY AT(22,1):"3 =>DA
RSTELLUNG DER F2 VON 1": :"K
ENNZAHL ?"
190 !CALL KEY(0,K,S):: IF S=
0 THEN 100
200 ACCEPT AT(24,11)VALIDATE
("1,2,3,4")BEEP SIZE(1):K
210 ON K GOSUB 220,500,750,2
000
220 CALL SCREEN(11)
230 A$="FF818181818181FF"
240 CALL CHAR(96,A$):: CALL
CHAR(104,A$):: CALL CHAR(112
,A$)
250 CALL COLOR(9,10,10,10,16
,16,11,16,10)
260 IMAGE #.#
270 CALL CLEAR
280 R,W,RW=0
290 FOR I=1 TO 400
300 RANDOMIZE
310 X=INT(2*RND)+1
320 Y=INT(2*RND)+1
330 IF X=Y AND X=1 THEN R=R+
1 ELSE IF X=Y AND X=2 THEN W
=W+1 ELSE RW=RW+1
340 F=MIN(R,W):: F=MIN(F,RW)
:: IF F=0 THEN F=.01
350 DISPLAY AT(1,1):"LAUFEND
E KREUZUNG :";I
360 DISPLAY AT(2,1):"=======
=====================
370 DISPLAY AT(4,1):"GENOTYP
EN :   VERHAELTNIS :"
380 DISPLAY AT(6,1)SIZE(-8):
"RR = ";R :: DISPLAY AT(6,25
)SIZE(-3):USING 260:R/F
390 DISPLAY AT(9,1)SIZE(-8):
"WW = ";W :: DISPLAY AT(9,25
)SIZE(-3):USING 260:W/F
400 DISPLAY AT(12,1)SIZE(-8)
:"RW = ";RW :: DISPLAY AT(12
,25)SIZE(-3):USING 260:RW/F
410 DISPLAY AT(15,1):"------
----------------------
420 DISPLAY AT(17,1):"PHAENO
TYP ROT   = ";R+RW
430 DISPLAY AT(20,1):"PHAENO
TYP WEISS = ";W
440 IF W=0 THEN 480
450 DISPLAY AT(24,1):"VERHAE
LTNIS WEISS/ROT=1/" :: V=(RR
+RW)/W :: DISPLAY AT(24,25)S
IZE(-3):USING 260:V
460 CALL HCHAR(7,3,96,R/15):
: CALL HCHAR(10,3,104,W/15):
: CALL HCHAR(13,3,112,RW/15)
470 CALL HCHAR(18,3,96,(R+RW
)/15):: CALL HCHAR(21,3,104,
W/15)
480 NEXT I
490 CALL KEY(0,K,S):: IF S=0
 THEN 490 ELSE 140
500 ! F2 DOMINATER ERBGANG M
IT 2 MERKMALEN
510 CALL CLEAR
520 F$="0006443C243C4200" ::
 F1$="0006443C3C3C4200" :: F
2$="00000000FF000000" :: F3$
="0101010101010101"
530 CALL CHAR(96,F$,104,F$,1
12,F1$,120,F1$,121,F2$,122,F
3$)
540 CALL COLOR(9,9,16,10,2,1
6,11,9,16,12,2,16)
550 CALL SCREEN(12)
560 DIM Q(4,4),ZAE(4,4)
570 Z1,Z2,Z3,Z4=0
580 FOR I=1 TO 4
```

```
590 FOR L=1 TO 4 :: ZAE(I,L)
=0
600 NEXT L
610 NEXT I
620 FOR I=1 TO 4 :: Q(1,I),Q
(I,1)=120 :: NEXT I :: Q(3,2
),Q(2,3)=120
630 Q(2,2),Q(4,2),Q(2,4)=104
 :: Q(3,3),Q(4,3),Q(3,4)=112
 :: Q(4,4)=96
640 FOR ZEI=1 TO 20
650 FOR SP=5 TO 25
660 RANDOMIZE
670 X=INT(4*RND)+1
680 Y=INT(4*RND)+1
685 ZAE(X,Y)=ZAE(X,Y)+1
690 CALL HCHAR(ZEI,SP,Q(X,Y)
,1)
700 IF Q(X,Y)=96 THEN Z1=Z1+
1 ELSE IF Q(X,Y)=112 THEN Z2
=Z2+1 ELSE IF Q(X,Y)=104 THE
N Z3=Z3+1 ELSE Z4=Z4+1
710 DISPLAY AT(21,1):"' = ";
Z1,"h = "¦Z3:"p = "¦Z2,"x =
"¦Z4
720 NEXT SP
730 NEXT ZEI
740 CALL KEY(0,K,S):: IF S=0
 THEN 740 ELSE 940
750 !DARSTELLUNG ::CALL CLEA
R :: CALL SCREEN(13)
760 CALL CHARPAT(42,A$):: CA
LL CHAR(96,A$):: CALL CHAR(1
04,A$):: CALL CHAR(112,A$)
770 CALL COLOR(9,10,1,10,16,
1,11,10,1)
780 FOR ZEI=1 TO 20
790 FOR SP=3 TO 27
800 RANDOMIZE
810 X=INT(2*RND)+1
820 Y=INT(2*RND)+1
830 IF X=Y AND X=1 THEN F=96
 ELSE IF X=Y AND X=2 THEN F=
104 ELSE F=112
840 IF F=96 THEN R=R+1 ELSE
IF F=104 THEN W=W+1 ELSE RW=
RW+1
850 CALL HCHAR(ZEI,SP,F,1)
860 NEXT SP
870 NEXT ZEI
880 CALL KEY(0,K,S):: IF S=0
 THEN 880
890 CALL COLOR(11,1,1)
900 FOR I=1 TO 50 :: NEXT I
910 CALL COLOR(11,2,1)
920 DISPLAY AT(23,1):"RR=";R
¦TAB(9)¦"WW="¦W¦TAB(18)¦"RW=
";RW
930 CALL KEY(0,K,S):: IF S=0
 THEN 890 ELSE 140
940 ! GENOTYPEN IN ZAHLEN
950 CALL CLEAR :: CALL SCREE
N(16)
960 DISPLAY AT(1,5):"GENOTYP
EN IN ZAHLEN"
970 DISPLAY AT(3,6):"SE   Se
   sE   se"
980 DISPLAY AT(6,1):"SE  SSE
E SSEe SsEE SsEe"
990 DISPLAY AT(11,1):"Se  SS
Ee SSee SsEe Ssee"
1000 DISPLAY AT(16,1):"sE  S
sEE SsEe ssEE ssEe"
1010 DISPLAY AT(21,1):"se  S
sEe Ssee ssEe ssee"
1015 N,M=1
1020 FOR Z=8 TO 23 STEP 5
1030 FOR SP=5 TO 20 STEP 5
1040 DISPLAY AT(Z,SP):ZAE(N,
M)
1045 M=M+1
1050 NEXT SP
1055 N=N+1 :: M=1
1060 NEXT Z
1070 FOR I=4 TO 24 STEP 5
1080 CALL HCHAR(I,5,121,22)
1090 NEXT I
1100 FOR I=6 TO 26 STEP 5
1110 CALL VCHAR(3,I,122,21)
1120 NEXT I
1130 CALL KEY(0,K,S):: IF S=
0 THEN 1130 ELSE 140
1140 CALL KEY(0,K,S):: IF S=
0 THEN 1140 ELSE 100
2000 END
```



DIE MENDEL'SCHEN REGELN

Dieses in Extended-Basic geschriebene Programm stellt eine Computersimulation, die den Verlauf von 400 Kreuzungen in der F2-Generation verdeutlicht.
Das Programm soll veranschaulichen, daß es sich bei den Mendel'schen Regeln nicht um feste Gesetze handelt, sondern, daß sich Populationen erst nach einer gewissen Anzahl von Kreuzungen den Mendel'schen Regeln nähern.
Wir hoffen nur, daß sich der Autor bei der Zusammenstellung seiner Algorithmen nicht geirrt hat, denn wir müssen zu unserer Schande eingestehen, daß wir von diesen Regeln herzlich wenig Ahnung besitzen. Doch einigen TI-Usern wird diese Simulation bestimmt von Nutzen sein können, und deshalb wollten wir auch auf einen Abdruck nicht verzichten.

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *     Speed-Racer      *
13 ! *                      *
14 ! *    Copyright  by     *
15 ! *                      *
16 ! *    Oliver  Storr     *
17 ! *                      *
18 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *      Joystick        *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     4746 Bytes       *
28 ! *                      *
29 ! ************************
30 !
190 GOTO 220
200 CALL SPRITE :: CALL DELS
PRITE :: CALL PATTERN :: CAL
L MOTION :: CALL VCHAR :: CA
LL COLOR :: CALL SCREEN :: C
ALL CLEAR :: A :: BB :: VFG
210 CALL HCHAR :: C :: CALL
KEY :: GHJ :: X :: Y
220 FOR I=65 TO 89 :: READ A
$ :: CALL CHAR(I,A$):: NEXT
I :: CALL CHAR(49,"ØØ1Ø1Ø1Ø1
Ø1Ø1Ø1Ø",50,"ØØD602Ø2D68Ø8ØD
6",51,"ØØD6Ø2Ø216Ø2Ø2D6")
230 CALL CHAR(52,"ØØ828282D6
Ø2Ø2Ø2",53,"ØØD68ØD6Ø2Ø2D6",
54,"ØØ8Ø8Ø8ØD68282D6",55,"ØØ
D6Ø2Ø2Ø2Ø2Ø2Ø2",56,"ØØD68282
D682D6")
240 ON WARNING NEXT :: CALL
CHAR(48,"ØØD68282828282D6"):
: CALL CHAR(57,"ØØD68282D602
Ø2Ø2"):: GOTO 290
250 DATA Ø03F666C746466EF,ØØ
7E313B3633337E,Ø03E634140603
F1E,Ø07E321119113F7E,Ø07F333
Ø3C38317F,Ø07F723239383078
260 DATA Ø03E6648504F663E,ØØ
7C603F3132327A03,Ø07C38181C1
A183C2,Ø03F13Ø6Ø6Ø6263F,Ø07B
72363C363673,Ø0783Ø3Ø3Ø72637
F,Ø07F2B2A49494963
270 DATA Ø07C76323A323477,ØØ
3E634559617F3E,Ø07E33313F3E3
Ø78,Ø03E634559617F3EØ7,Ø07F3
13A34323179,Ø03E413C06634F3E
,Ø07F492AØ8Ø8Ø81C
280 DATA Ø06F2626666E7E37,ØØ
73622232321EØC,ØØ7722224149 3
6,Ø0793214Ø814264F,Ø07341623
418183CØØ7F4326ØC1A317F
290 CALL CLEAR :: CALL SCREE
N(2):: FOR I=Ø TO 14 :: CALL
 COLOR(I,2,2):: NEXT I
300 PRINT ".ooo..ooo..oooo.o
ooo.ooo....a.....a..a.a....a
....a..a...o.....o..o.o....o
....o..o....aa...aaaa.aaa..a
aa..a..a......o..o....o....o
....o..o.."
310 PRINT "a..a..a....a....a
....a..a....oo...o....oooo.o
ooo.ooo": : :
320 PRINT "aaaa...aa...aaa..
aaaa.aaaa..o...o.o..o.o...o.
o....o...o.a...a.a..a.a.....
a....a...a.oooo..oooo.o.....
ooo..oooo..a.a...a..a.a.....
a....a.a."
330 PRINT "o..o..o..o.o...o.
o....o..o..a...a.a..a..aaa..
aaaa.a...a"
340 PRINT : : : : :
350 FOR I=1 TO 14 :: CALL CO
LOR(I,11,1):: NEXT I
360 DISPLAY AT(24,4):"(C)198
6 STORR-SOFT"
370 CALL COLOR(10,5,5,9,7,7)
380 DISPLAY AT(21,1):"LENKGE
SCHWINDIGKEIT(5-15)" :: ACCE
PT AT(21,26)BEEP VALIDATE(DI
GIT):P1 :: IF P1<5 OR P1>15
THEN 380
390 CALL SOUND(150,1400,5)::
 DISPLAY AT(22,2):"ZUM START
 BUTTON DRUECKEN"
400 FOR AQ=1 TO 2 :: CALL KE
Y(AQ,H,J):: IF H=18 THEN 420
410 NEXT AQ :: GOTO 400
420 CALL CLEAR :: A=20 :: RA
NDOMIZE :: CALL SCREEN(5)::
B=Ø
430 CALL CHAR(91,"Ø3Ø3Ø3Ø3Ø3
Ø3Ø3Ø3C0C0C0C0C0C0C0C0E0E0E0
E0E0E0E0E0E0"):: ! ZEICHEN FUE
R BILDSCHIRMAUFBAU
440 CALL CHAR(96,RPT$("Ø3",8
)&RPT$("Ø",48)):: ! LEITPFOS
TEN
450 CALL CHAR(124,"1FØ31B1F1
BØ3Ø3Ø3Ø3Ø33B3B3F3B3BØ7F8CØD
8F8D8C0C0C0C0C0DCDCFCDCDCEØ"
):: ! AUTO VORWAERTS
460 CALL CHAR(120,"1FØ31B3F3
3Ø3Ø3Ø3Ø3Ø33B3B3F3B3BØ7F8CØC
CF8D8C0C0C0C0C0DCDCFCDCDCEØ"
):: ! REIFEN RECHTS
470 CALL CHAR(116,"1F03333F1
BØ3Ø3Ø3Ø3Ø33B3B3F3B3BØ7F8CØD
8FCCCC0C0C0C0C0DCDCFCDCDCEØ"
):: ! REIFEN LINKS
480 CALL CHAR(128,"Ø73B3B3F3
B3BØ3Ø3Ø3Ø3Ø31B1F1BØ31FEØDCD
CFCDCDCC0C0C0C0C0D8F8D8C0F8"
):: !  AUTO RUECKWAERTS
490 CALL CHAR(132,"ØØØØØØB8B
89ØFFFFFFFF9ØB8B8ØØØØØØØØØ3
E3E3EØ9FFFFFFFFØ93E3E3EØØØØ"
```

```
):: ! AUTO LINKS
500 CALL CHAR(136,"0000C7C7C7
C90FFFFFFFF907C7C7C000000000
01D1D09FFFFFFFF091D1D000000"
):: ! AUTO RECHTS
510 FOR I=1 TO 14 :: CALL CO
LOR(I,5,5):: NEXT I
520 CALL HCHAR(1,3,32,768)::
 CALL HCHAR(23,1,112,64):: C
ALL VCHAR(3,4,91,20):: CALL
VCHAR(3,13,92,20):: CALL VCH
AR(3,16,93,20)
530 CALL HCHAR(1,1,112,64)
540 FOR I=5 TO 12 :: CALL VC
HAR(3,I,104,20):: NEXT I ::
CALL VCHAR(3,1,112,20):: CAL
L VCHAR(3,32,112,20)
550 CALL COLOR(1,4,4,11,5,5,
8,15,4,10,15,15)
560 FOR I=0 TO 7 :: CALL COL
OR(I,2,4):: NEXT I
570 DISPLAY AT(19,17):"(C)19
86" :: DISPLAY AT(21,15):"ST
ORR-SOFT"
580 DISPLAY AT(6,15):"SPEED:
";A*3+50 :: DISPLAY AT(9,15)
:"AUTOS:";BB
590 CALL SPRITE(#3,96,5,120,
18,#4,96,5,120,100)
600 A=20 :: BB=0 :: CALL SPR
ITE(#1,124,9,130,65):: CALL
MAGNIFY(3)
610 CALL SOUND(10,110+A*2,0,
-7,8):: CALL BEWEGUNG(A,P1,A
Q):: DISPLAY AT(6,21):A*3+50
 :: DISPLAY AT(9,21):BB
620 CALL COINC(ALL,J):: IF J
 THEN 660
630 CALL FEIND(A,VFG,BB):: C
ALL POSITION(#1,X1,Y1):: IF
Y1>82 OR Y1<30 THEN 660
640 CALL GUCK(A,VFG):: CALL
COINC(ALL,GHJ)
650 IF NOT GHJ THEN 610
660 J=124 :: CALL MOTION(#1,
0,0,#2,0,0,#3,0,0,#4,0,0)::
FOR S=0 TO 30 :: CALL PATTER
N(#1,J):: IF J=136 THEN J=12
4 ELSE J=J+4
670 CALL SOUND(-500,-6,30-S,
170-2*S,30-S):: NEXT S :: FO
R S=0 TO 30
680 CALL SOUND(-500,-5,S)
690 NEXT S :: CALL DELSPRITE
(#1,#2):: DISPLAY AT(11,16):
"NOCHMAL? J/N" :: CALL SOUND
(150,1400,5)
700 CALL KEY(AQ,M1,L1):: CAL
L KEY(0,M2,L2):: IF M1=18 OR
 L2<>0 THEN 710 ELSE 700
710 IF M1=18 OR M2=74 OR M2=
106 THEN 740 ELSE IF M2=110
OR M2=78 THEN 750 ELSE 730
720 GOTO 740
730 FOR Z=1 TO 50 :: NEXT Z
:: CALL SOUND(150,140,5):: G
OTO 700
740 BG=0 :: DISPLAY AT(11,16
):"............." :: GOTO 60
0
750 CALL CLEAR :: END
760 SUB BEWEGUNG(A,P1,AQ)::
CALL JOYST(AQ,X,Y)
770 IF X=4 OR X=4 AND Y=4 TH
EN CALL PATTERN(#1,120):: CA
LL MOTION(#1,0,P1)
780 IF X=-4 OR X=-4 AND Y=4
THEN CALL PATTERN(#1,116)::
CALL MOTION(#1,0,-P1)
790 IF X=0 THEN CALL PATTERN
(#1,124):: CALL MOTION(#1,0,
0)
800 IF Y=4 AND A<127 THEN A=
A+1
810 IF Y=-4 AND A>21 THEN A=
A-5
820 SUBEND
830 SUB FEIND(A,VFG,BB):: IF
 VFG=1 THEN SUBEXIT
840 C=INT(RND*(83-35+1)+35)
850 BB=BB+1 :: CALL SPRITE(#
2,124,5,1,C,A,0)
860 SUBEND
870 SUB GUCK(A,VFG):: CALL M
OTION(#3,A,0,#4,A,0)
880 CALL MOTION(#2,A,0):: CA
LL POSITION(#2,Z,C):: IF Z>1
77 THEN CALL DELSPRITE(#2)::
 VFG=0 ELSE VFG=1
890 SUBEND
900 !
```

SPEED-RACER

Wollten Sie nicht schon immer Rennfahrer werden? Jetzt wird Ihr Wunschtraum möglich! Sie benötigen nur die TI 99/4A-Konsole, Ext. Basic, Joystick und dieses Listing. Nachdem Sie den Befehl 'RUN' eingetippt haben, haben Sie nur noch wenige Sekunden zum Anschnallen Zeit. Vor dem Start werden Sie nach der Lenkgeschwindigkeit gefragt, wobei 5 für den Anfänger und 15 für den Profi gedacht ist. Danach können Sie Ihren Joystick wählen, indem Sie den dazugehörigen roten Button drücken. Spätestens jetzt, im Cockpit, wird Ihnen die Schönschrift auffallen, in dem die überholten Autos gezählt werden, die den Punkten gleichkommen.
Sie können beschleunigen, indem Sie den Joystick nach vorn drücken, umgekehrt machen Sie von der schnellwirkenden Bremse Gebrauch. Zum Lenken bewegt man den Steuerknüppel in die gewünschte Richtung, wobei man den Joystick auch diagonal bewegen kann. Die Geschwindigkeit ist unbegrenzt, da macht das Gasgeben so richtig Spaß!!! Nach einem 'Crash' kann man durch Joystick oder Tastendruck neu starten.
Gu(r)te Fahrt und viel Speed!!!

WAY TO HOME

Way to Home ist ein im Extended-Basic geschriebenes Grafik- und Textspiel, wobei alle Eingaben klein geschrieben werden müssen. Nach dem Titelbild und der Titelmusik geht es los. Es erscheint ein Bild mit einem Raum und einigen Angaben über diesen Raum, die im unteren Bildteil stehen. Der Computer fragt dann: „Was soll ich tun?" Nun gibt man das ein, was der Computer tun soll. Der Befehl lautet „geh (eine der vier Richtungen)", z.B. „geh Norden", wenn nicht auf andere Befehle verwiesen wird. Ziel des Spiels ist es, das „Home" zu finden. Aber Vorsicht, zwei Räume sind nicht ganz „normal". Tip: Man achte auf die Angabe „Ausgang:" und suche den Eingang zum Geheimgang.
Ein Tip noch zum Programmieren: Wenn einige Zeilen zu lang sein sollten, bzw. der Computer am Ende einer Zeile piepst und man kann nicht weiterschreiben, dann schließe man die Zeile ordnungsgemäß ab und schicke sie mit "ENTER" in den Computer. Danach drücke man FCTN-8 und die Zeilensperre ist aufgehoben, man kann jetzt am Ende der Zeile weiterschreiben.

COMMANDER

Das Spiel 'Commander' ist für zwei User geschrieben. Im Spiel haben sie die Namen Stoom und Metu und müssen ihre Fähigkeiten im Kampf unter Beweis stellen. Nach dem Start erscheint das Innere eines Cockpits. Zuerst muß Commander Stoom zeigen, was er 'drauf' hat. Um anzufangen, wird Fire gedrückt. Jetzt erscheint ein feindlicher Jäger und der Spieler hat 10 Sekunden Zeit, ihn abzuschießen. (Über die Steuerung mit Joystick 1 sei nichts gesagt. Wie heißt es doch so schön: Probieren geht über Studieren!) Zu beachten ist nur, daß man 40 Schuß zur Verfügung hat und 'Fuel' auch begrenzt ist. Schafft man es aber, 3 Schiffe innerhalb von jeweils 5 Sekunden zu vernichten, so füllt sich der Energievorrat wieder auf. Commander Metu kommt dran, wenn Stoom entweder alle Schüsse verbraucht hat, 'Fuel' verbraucht ist oder seine 3 Leben zu Ende sind (man verliert immer dann eines, wenn man es nicht schafft, das Limit von 10 Sekunden einzuhalten). Im 2. Bild kämpfen Stoom und Metu gegeneinander. Die beiden haben sich in der weiten Ebene auf einem weit entfernten Planeten mit ihren Kampfjägern verabredet und liefern sich nun ein heißes Duell. Schüsse können nur horizontal abgefeuert werden. Jeder Treffer kostet den Getroffenen Energie. Derjenige, dessen Energie zuerst verbraucht ist, hat verloren.
Jetzt folgt noch eine Anzeige der Punkteverteilung. Ich wünsche viel Spaß bei diesem Abenteuer in weiter Zukunft!
Auf ein Titelbild wurde aufgrund des Speichers verzichtet. Wer einen Sprachsynthesizer besitzt, kann diesen zur sprachlichen Untermalung des Spieles anschließen. Der Computer fragt nach RUN automatisch den Userport ab!

Dietmar Augustin

LISTINGS

```
100 REM ******************
110 REM *   WAY TO HOME   *
120 REM *================*
130 REM *     (c) by     *
140 REM *     daniel     *
150 REM *    eikelmann   *
160 REM *================*
170 REM * benoe. geraete: *
180 REM *                *
190 REM * TI99/4A KONSOLE *
200 REM *    EXT. BASIC  *
210 REM *   CASSETTENREC. *
220 REM *                *
230 REM * SPEICHERBELEG.: *
240 REM * ca. 12040 BYTES *
250 REM ******************
252 REM
254 REM Die Punkte im Prog-
        ramm sind natuerlich
        wieder nur als Leer-
        zeichen zu betrach-
        ten.
255 REM Also nicht mit abtip
        pen !!!
257 REM
260 ON BREAK NEXT
270 GOTO 380
280 ACCEPT AT(24,2)SIZE(-19)
BEEP VALIDATE(" abcdefghijkl
mnopqrstuvwxyz"):A$
290 RETURN
300 GOTO 380
310 CALL CHAR(96,"FFFF000000
000000",123,"181818FFFF18181
8",124,"0038440408100010")
320 CALL CHAR(125,"0000303000
0303000",121,"4040404040404040
40")
330 CALL CHAR(106,"000000FF0
0000000",112,"0000000000301020
20"):: FOR I=9 TO 12 :: CALL
 COLOR(I,16,2):: NEXT I
340 DISPLAY AT(15,1):"''''''
''''''''''''''''''''''' ausga
ng}...........y...n.........
..............y........jjjjj
jjjjjjjjjjjjjy w { o "
350 DISPLAY AT(19,1):" du bi
st im}........y.............
..............y...s....jjjjj
jjjjjjjjjjjjjy........was s
oll ich tun!..y......."
360 DISPLAY AT(23,1):"......
..............y.............
..............y......."
370 RETURN
380 CALL MAGNIFY(2)
390 CALL CLEAR :: CALL SCREE
N(2)
400 RESTORE 1950 :: FOR I=1
TO 9 :: READ ZZ,ZC$,A,FC,ZE,
SP :: CALL CHAR(ZZ,ZC$):: CA
LL SPRITE(#A,ZZ,FC,ZE,SP)::
NEXT I
410 FOR F=9 TO 12 :: CALL CO
LOR(F,16,1):: NEXT F
420 CALL COLOR(2,16,2):: CAL
L CHAR(40,"0000004428101010"
)
430 CALL CHAR(120,"3C4299A1A
199423C"):: DISPLAY AT(24,4)
:" x b( daniel eikelmann "
440 RESTORE 1980 :: FOR I=1
TO 32 :: READ D,F :: CALL SO
UND(D,F,8):: NEXT I :: FOR I
=1 TO 500 :: NEXT I
450 CALL DELSPRITE(ALL):: CA
LL CHARSET
460 RANDOMIZE
470 Z=INT(RND*7)+1
480 ON Z GOTO 490,610,720,82
0,920,1020,1120,1220
490 CALL CHAR(33,"FFFFFFFFFF
FFFFFF",40,"FFFFFFFFFFFFFFFF
",48,"FFFFFFFFFFFFFFFF")
500 CALL CHAR(42,"E7E7E7E7E7
E7E7E7")
510 CALL COLOR(1,11,2,2,7,2,
3,16,2)
520 FOR F=9 TO 12 :: CALL CO
LOR(F,16,2):: NEXT F
530 DISPLAY AT(1,1):"!!!!!!!
!!!!!!!!!!!!!!!!!!!!..(*(..
.......(...............(*(..
.....(((((.............(*(..
.......(.............."
540 DISPLAY AT(5,1):"..(*(..
((((...(...((((........(*(.(
(((((..(..((((((.....!!!!!!(
00000(!!!!(00000(!!!!!!!!!!(
(00000(!!!((00000(!!!"
550 DISPLAY AT(9,1):"!!!!!!(
!(00000(!!(!(00000(!!!!!!!!!
!!((((((((!!!!((((((!!!!!!!!
!!(!!!!!(!!!!(!!!!!(!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!"
560 X=11 :: FOR I=1 TO 3 ::
X=X+1 :: DISPLAY AT(X,1):"!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!"
:: NEXT I
570 GOSUB 310
580 DISPLAY AT(16,10)SIZE(-1
0):"westenp" :: DISPLAY AT(1
7,1)SIZE(-20):" osten" :: DI
SPLAY AT(20,1)SIZE(-20):" sc
hlafzimmer"
590 GOSUB 280
600 IF A$="geh westen" THEN
820 :: IF A$="geh osten" THE
N 720 :: GOTO 590
610 CALL CHAR(33,"FFFFFFFFFF
FFFFFF",40,"FFFFEFEFEFFFFFFF
",41,"FFFFFFFFFFFFFFFF",48,"
FFFFFFFFFFFFFFFF")
620 CALL COLOR(1,16,1,2,15,2
,3,7,1):: FOR I=9 TO 12 :: C
```

```
ALL COLOR(I,16,1):: NEXT I
630 DISPLAY AT(1,1):"ØØØØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØØ))))
ØØØØØØØØØ)))ØØØ))))ØØØØØ()))
ØØØØ)))ØØØØØØØØ()))ØØØØØØØØØ
ØØØØ)))ØØØØØØØØØØØØØØ"
640 DISPLAY AT(5,1):"ØØØØØØØ
ØØØØ())ØØØØØØØØØØØØØØ))))))
))))))))))))))))))))))))()))())
)())()())()))()))()))))))))))
))))))))))))))))))))"
650 X=9 :: FOR I=1 TO 6 :: D
ISPLAY AT(X,1):"!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!" :: X=X+1 :
: NEXT I
660 GOSUB 310
670 DISPLAY AT(16,10)SIZE(-1
0):"osten"
680 DISPLAY AT(17,1)SIZE(-20
):" sueden"
690 DISPLAY AT(20,1)SIZE(-20
):" kochraum"
700 GOSUB 280
710 IF A$="geh osten" THEN 8
20 :: IF A$="geh sueden" THE
N 920 :: GOTO 700
720 CALL CHAR(56,"FFFFFFFFFF
FFFFFF",40,"FFFFFFFFFFFFFFFF
",41,"FFFF9F9F9FFFFFFF")
730 CALL CHAR(48,"183C7E7E7E
424242",49,"4242424242ØØØØØØ
",33,"FFFFFFFFFFFFFFFF",64,"
FFFFFFFFFFFFFFFF")
740 CALL COLOR(4,16,1,2,5,16
,3,5,8,1,15,1,5,8,1):: FOR I
=9 TO 12 :: CALL COLOR(I,16,
1):: NEXT I
750 DISPLAY AT(1,1):"@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@8888@@@@@@@@@@@@@@@@@@@@@@
@@@8888@@@@@@@@@@@@(((((@@@@
@@@@@@@@@@@@@@@@@@@@)("
760 DISPLAY AT(5,1):")((@@@@
ØØØ((((@@@@@@@@@@@@@@(((((@@@@
11@@((@@@@((((((((((((((((@@@@
@@@@@@@@@@((((((((((((!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!"
770 X=9 :: FOR I=1 TO 6 :: D
ISPLAY AT(X,1):"!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!" :: X=X+1 :
: NEXT I :: GOSUB 310
780 DISPLAY AT(16,10)SIZE(-1
0):"westenp"
790 DISPLAY AT(17,1)SIZE(-20
):" sueden" :: DISPLAY AT(20
,1)SIZE(-20):" badezimmer"
800 GOSUB 280
810 IF A$="geh westen" THEN
490 :: IF A$="geh sueden" TH
EN 1230 :: GOTO 800
820 CALL CHAR(33,"ØØ7777ØØØØ
DDDDØØ",40,"FFFFFFFFFFFFFFFF
",41,"FFFFCFCFCFCFFFFF")
830 CALL CHAR(48,"FFFFFFFFFF
FFFFFF",49,"FFFFCFCFCFCFFFFF
",56,"FFFFFFFFFFFFFFFF")
840 CALL COLOR(1,16,2,2,10,2
,3,7,2):: FOR I=9 TO 12 :: C
ALL COLOR(I,16,1):: NEXT I
850 DISPLAY AT(1,1):"!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!ØØØØØØØØØ!!!!!!!
!!!!!!!!!!!Ø!Ø!Ø!Ø!Ø"
860 DISPLAY AT(5,1):"!!!!!!!
!!!!!!ØØØØ!!Ø!!!!!!!Ø!!!!!!!
!!!!!!1ØØØ!!Ø!!!!!!!Ø!!!!!!!
!!!!!!ØØØØ!!Ø!!!!!!!Ø'''''''
''''''''''''''''''''''''"
870 X=9 :: FOR I=1 TO 6 :: D
ISPLAY AT(X,1):"............
................" :: X=X+1 :
: NEXT I :: GOSUB 310
880 DISPLAY AT(16,10)SIZE(-1
0):"westenp"
890 DISPLAY AT(17,1)SIZE(-20
):" ostenpsueden" :: DISPLAY
 AT(20,1)SIZE(-20):" flur"
900 GOSUB 280
910 IF A$="geh westen" THEN
610 :: IF A$="geh osten" THE
N 490 :: IF A$="geh sueden"
THEN 1020 :: GOTO 900
920 CALL CHAR(33,"F7B5ADEBFA
AE6D5D",40,"FFFFFFFFFFFFFFFF
",41,"FFFFFCFCFCFCFFFF")
930 CALL CHAR(56,"FFFFFFFFFF
FFFFFF",64,"FFFFFFFFFFFFFFFF
")
940 CALL COLOR(1,11,2,2,10,2
,4,7,1,5,12,1):: FOR I=9 TO
12 :: CALL COLOR(I,16,1):: N
EXT I
950 DISPLAY AT(1,1):"!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!(((
(!!!!!!!(((((((((!!!!!!!!!!!(((
(!!!!!!!)(((((((!!!!!!!!!!!)((
(!!!!!!!(((((((!!!!!!!"
960 DISPLAY AT(5,1):"!!!!!(((
(!!!!!!!!!!!!!!!!!!!!!!!!!!!(((
(!!!!8!!!!!!!!!!8!!!!!!!!!(((
(!!!!8!!888888!!8!!!!@@@@@@@
@@@@@@888@@@@@@@@@@@@@@"
970 X=10
980 DISPLAY AT(9,1):"@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@" :: FO
R I=1 TO 5 :: DISPLAY AT(X,1
):"@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@" :: X=X+1 :: NEXT I
990 GOSUB 310 :: DISPLAY AT(
16,10)SIZE(-10):"nordenp"
1000 DISPLAY AT(17,1)SIZE(-2
0):" osten" :: DISPLAY AT(20
,1)SIZE(-20):" esszimmer"
1010 GOSUB 280 :: IF A$="geh
 norden" THEN 610 :: IF A$="
```

```
geh osten" THEN 1Ø2Ø :: GOTO
 1Ø1Ø
1Ø2Ø CALL CHAR(33,"F7B5ADEBF
AAE6D5D",4Ø,"FFFFFFFFFFFFFFF
F",41,"FFFFFCFCFCFCFFFF")
1Ø3Ø CALL CHAR(56,"FFFFFFFFF
FFFFFFF",64,"FFFFFFFFFFFFFFF
F")
1Ø4Ø CALL COLOR(1,11,2,2,1Ø,
2,4,7,1,5,12,1):: FOR I=9 TO
 12 :: CALL COLOR(I,16,1)::
NEXT I
1Ø5Ø DISPLAY AT(1,1):"!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!((((
!!!!!!!!((((((((!!!!!!!!((((
!!!!!!!!)((((((((!!!!!!!!)(((
!!!!!!!!((((((((!!!!!!!"
1Ø6Ø DISPLAY AT(5,1):"!!((((
!!!!!!!!!!!!!!!!!!!!!!!!!!((((
!!!!!8!!!!!!!!!!!!8!!!!!((((
!!!!!8!!!!!!!!!!!!8!!!@@@@@@
@@@@@888@888888@888@@@"
1Ø7Ø X=1Ø
1Ø8Ø DISPLAY AT(9,1):"@@@@@@
@@@@@888@@8@@8@@888@@@" :: F
OR I=1 TO 5 :: DISPLAY AT(X,
1):"@@@@@@@@@@@@@@@@@@@@@@@@
@@@@" :: X=X+1 :: NEXT I
1Ø9Ø GOSUB 31Ø :: DISPLAY AT
(16,1Ø)SIZE(-1Ø):"nordenp"
11ØØ DISPLAY AT(17,1)SIZE(-2
Ø):" westenposten" :: DISPLA
Y AT(2Ø,1)SIZE(-2Ø):" wohnzi
mmer"
111Ø GOSUB 28Ø :: IF A$="geh
 westen" THEN 92Ø :: IF A$="
geh norden" THEN 82Ø :: IF A
$="geh osten" THEN 112Ø :: G
OTO 111Ø
112Ø CALL CHAR(33,"ØØ7777ØØØ
ØDDDDØØ")
113Ø CALL CHAR(48,"FFFFFFFFF
FFFFFFF",49,"FFFFCFCFCFCFFFF
F",56,"FFFFFFFFFFFFFFFF")
114Ø CALL COLOR(1,16,2,2,1Ø,
2,3,7,2):: FOR I=9 TO 12 ::
CALL COLOR(I,16,1):: NEXT I
115Ø DISPLAY AT(1,1):"!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!Ø!Ø!Ø!!!!!!!!!!
!!!!!!!!!!!!!ØØØØØ!!!!"
116Ø DISPLAY AT(5,1):"!!ØØØØ
!!!!!!!ØØØØ!!!!Ø!!!!!!!!1ØØØ
!!!!!!!1ØØØ!!!!Ø!!!!!!!!ØØØØ
!!!!!!!ØØØØ!!!ØØØ!!!!!''''''
''''''''''''''''''''''''"
117Ø X=9 :: FOR I=1 TO 6 ::
DISPLAY AT(X,1):"...........
..................." :: X=X+1
:: NEXT I
118Ø GOSUB 31Ø :: DISPLAY AT
(16,1Ø)SIZE(-1Ø):"westenp"
```

LISTINGS

```
119Ø DISPLAY AT(17,1)SIZE(-2
Ø):" osten" :: DISPLAY AT(2Ø
,1)SIZE(-2Ø):" flur"
12ØØ GOSUB 28Ø
121Ø IF A$="geh westen" THEN
 1Ø2Ø :: IF A$="geh osten" T
HEN 122Ø :: GOTO 12ØØ
122Ø CALL CHAR(4Ø,"FFFFFFFFF
FFFFFFF"):: CALL CHAR(41,"FF
FFFCFCFCFCFFFF")
123Ø CALL CHAR(49,"FFFFFFFFF
FFFFFFF",56,"ØØ7777ØØØØDDDDØ
Ø")
124Ø CALL COLOR(4,16,2,2,1Ø,
2,3,16,2):: FOR I=9 TO 12 ::
 CALL COLOR(I,16,1):: NEXT I
125Ø DISPLAY AT(1,1):"888888
888888888888888888888888888((
((888888111111188888888888((
((888888111111188888888888)(
((88888811111118888888"
126Ø DISPLAY AT(5,1):"8888((
((888888111111188((((88888((
((888888111111188)(((88888((
((888888111111188((((8''''''
''''''''''''''''''''''''"
127Ø X=9 :: FOR I=1 TO 6 ::
DISPLAY AT(X,1):"...........
..................." :: X=X+1
:: NEXT I
128Ø GOSUB 31Ø :: DISPLAY AT
(16,1Ø)SIZE(-1Ø):"westenp"
129Ø DISPLAY AT(17,1)SIZE(-2
Ø):" nordenp|" :: DISPLAY AT
(2Ø,1)SIZE(-2Ø):" flur"
13ØØ GOSUB 28Ø
131Ø IF A$="geh westen" THEN
 112Ø :: IF A$="geh norden"
THEN 72Ø :: IF A$="geh suede
n" THEN 132Ø :: GOTO 13ØØ
132Ø CALL CHAR(56,"ØØ7777ØØØ
ØDDDDØØ")
133Ø CALL COLOR(4,7,2):: FOR
 I=9 TO 12 :: CALL COLOR(I,1
6,1):: NEXT I
134Ø X=1 :: FOR I=1 TO 7 ::
X=X+1 :: DISPLAY AT(X,1):"88
888888888888888888888888888"
:: NEXT I
135Ø DISPLAY AT(8,1):"''''''
''''''''''''''''''''''''"
136Ø X=9 :: FOR I=1 TO 6 ::
DISPLAY AT(X,1):"...........
..................." :: X=X+1
:: NEXT I
137Ø GOSUB 31Ø :: DISPLAY AT
(16,1Ø)SIZE(-1Ø):"|p|p|"
138Ø DISPLAY AT(17,1)SIZE(-2
Ø):"......." :: DISPLAY AT(2
Ø,1)SIZE(-2Ø):" geheimgang"
139Ø GOSUB 28Ø
14ØØ IF A$="geh norden" THEN
 122Ø :: IF A$="geh westen"
```

```
THEN 1420 :: IF A$="geh sued
en" THEN 1520 :: GOTO 1390
1410 REM
1420 GOSUB 310 :: DISPLAY AT
(16,10)SIZE(-10):"|p|"
1430 DISPLAY AT(17,1)SIZE(-2
0):"........" :: DISPLAY AT(
20,1)SIZE(-20):" geheimgang"
1440 GOSUB 280
1450 IF A$="geh osten" THEN
1320 :: IF A$="geh sueden" T
HEN 1470 :: GOTO 1440
1460 REM ******************
1470 GOSUB 310 :: DISPLAY AT
(16,10)SIZE(-10):"|p|"
1480 DISPLAY AT(17,1)SIZE(-2
0):"........" :: DISPLAY AT(
20,1)SIZE(-20):" geheimgang"
1490 GOSUB 280
1500 IF A$="geh norden" THEN
 1420 :: IF A$="geh westen"
THEN 1760 :: GOTO 1490
1510 REM ******************
1520 GOSUB 310 :: DISPLAY AT
(16,10)SIZE(-10):"...denp"
1530 DISPLAY AT(17,1)SIZE(-2
0):" no..    " :: DISPLAY AT(
20,1)SIZE(-20):" geheimgang"
1540 GOSUB 280
1550 IF A$="geh norden" THEN
 1320 :: IF A$="geh sueden"
THEN 1570 :: GOTO 1540
1560 REM ******************
1570 GOSUB 310 :: DISPLAY AT
(16,10)SIZE(-10):"|p....en"
1580 DISPLAY AT(17,1)SIZE(-2
0):"........" :: DISPLAY AT(
20,1)SIZE(-20):" geheimgang"
1590 GOSUB 280
1600 IF A$="geh norden" THEN
 1520 :: IF A$="geh westen"
THEN 1620 :: GOTO 1590
1610 REM ******************
1620 CALL CHAR(40,"FFFFFFFFF
FFFFFFF",48,"FFFFFFFFFFFFFFF
F",56,"FFFFFFFFFFFFFFFF")
1630 CALL CHAR(120,"0000040B
1020400",113,"0010101010100
010",112,"0000007844784040",
106,"0000000808084830")
1640 CALL CLEAR
1650 CALL SCREEN(2):: CALL C
OLOR(2,11,2,3,6,2,4,7,1):: F
OR I=9 TO 12 :: CALL COLOR(I
,16,1):: NEXT I
1660 DISPLAY AT(2,1):"..((((
((((((((((((((((((....(000
0000000000000000000(....(080
0808888080000808888(....(080
0808000808808080000("
1670 DISPLAY AT(6,1):"..(088
8808000808080808888(....(080
0808080800008080000(....(080
0808888080000808888(....(000
0000000000000000000("
1680 DISPLAY AT(10,1):"..(((
((((((((((((((((((((..."
1690 DISPLAY AT(13,1):"...si
e haben es geschafft.....
.....,...d
en weg nach hause"
1700 DISPLAY AT(17,10):"zu f
indenq"
1710 FOR I=1 TO 100 :: CALL
SOUND(-20,262,0):: CALL SOUN
D(-20,330,2):: NEXT I
1720 DISPLAY AT(20,5):"neues
 spiel! jxn"
1730 ACCEPT AT(22,10)SIZE(-4
)VALIDATE(" neija"):B$
1740 IF B$="ja" THEN 100 ::
IF B$="nein" THEN 2020 :: GO
TO 1730
1750 REM ******************
1760 GOSUB 310 :: DISPLAY AT
(16,10)SIZE(-10):"|p|p|"
1770 DISPLAY AT(17,1)SIZE(-2
0):"........" :: DISPLAY AT
(20,1)SIZE(-20):" geheimgang
"
1780 GOSUB 280
1790 IF A$="geh norden" THEN
 1810 :: IF A$="geh osten" T
HEN 1470 :: IF A$="geh suede
n" THEN 1870 :: GOTO 1780
1800 REM ******************
1810 DISPLAY AT(15,1):"''''''
'''''''''''''''''''''''''' dies
er raum hat....y...n..."
1820 DISPLAY AT(17,1)SIZE(-2
0):" keinen ausgang. man" ::
 DISPLAY AT(18,1)SIZE(-20):"
 braucht einen vers" :: DISP
LAY AT(19,1)SIZE(-20):" aus
alibaba und die"
1830 DISPLAY AT(20,1)SIZE(-2
0):" vierzig raeuber." :: DI
SPLAY AT(21,1)SIZE(-20):" hi
lfe}....dich "
1840 GOSUB 280
1850 IF A$="sesam oeffne dic
h" THEN 1760 :: GOTO 1840
1860 REM ******************
1870 GOSUB 310 :: DISPLAY AT
(16,10)SIZE(-10):"|p|"
1880 DISPLAY AT(17,1)SIZE(-2
0):"........" :: DISPLAY AT(
20,1)SIZE(-20):" geheimgang"
1890 GOSUB 280
1900 IF A$="geh westen" THEN
 1920 :: IF A$="geh norden"
THEN 1760 :: GOTO 1890
1910 REM ******************
1920 FOR I=1 TO 24 :: X=X+1
:: DISPLAY AT(X,1):"........
......................" :: NEX
```

```
T I :: DISPLAY AT(6,1):"du b
ist leider durch eine..."
1930 DISPLAY AT(7,1):"luke g
efallenpkannst nicht " :: DI
SPLAY AT(8,1):"zurueck und l
andest im.....sclafzimmer."
1940 FOR I=1 TO 2000 :: NEXT
 I :: GOTO 490
1950 DATA 33,8181819999995A3
C,1,16,25,25,34,3C428181FF81
8181,2,16,25,55,35,8181423C1
8181818,3,16,25,85
1960 DATA 36,FF1818181818181
8,4,16,85,85,37,FF8181818181
81FF,5,16,85,115,38,818181FF
FF818181,6,16,145,115
1970 DATA 39,FF818181818181F
F,7,16,145,145,42,81C3A59999
818181,8,16,145,175,41,FF808
0FF8080808FF,9,16,145,205
1980 DATA 125,220,125,247,25
0,196,125,220,125,247,250,19
6,125,220,125,247,125,262,12
5,247
1990 DATA 125,220,125,247,25
0,196,125,220,125,247,250,19
6,125,220,125,247,250,196
2000 DATA 125,220,125,247,12
5,262,125,247,125,220,125,24
7,125,196,125,247
2010 DATA 250,220,125,196,12
5,247,250,220,250,196
2020 END
```

LISTINGS

```
100 !  ********************
110 ! *******TI-99/4A*******
120 ! **                   **
130 ! *T    COMMANDER      T*
140 ! *I                   I*
150 ! *--------------------*
160 ! *9  copyright 1986  9*
170 ! *9  by  D.Augustin  9*
180 ! */------------------/*
190 ! *4     TI-99/4A      4*
200 ! *A  Extended BASIC   A*
210 ! **    Joystick 1     **
220 ! **                   **
230 ! *******TI-99/4A*******
240 !  ********************
250 !
260 CALL PEEK(-28672,Z):: IF
 Z=0 THEN 270 ELSE CALL SAY(
"PLEASE+WAIT")
270 RANDOMIZE :: OPTION BASE
 1 :: CALL CLEAR :: CALL MAG
NIFY(3):: CALL B2
280 DIM R(2),B(2),P(2),A$(6)
290 R(1),R(2)=40 :: SI=1 ::
T=1000 :: B(1),B(2)=7 :: NA$
="EKK@KKIKKJ" :: W,P(1),P(2)
=0
300 DEF ZE=INT(RND*96)+40
310 DEF SP=INT(RND*192)+40
320 DEF ZE1=INT(RND*10)-5
330 DEF SP1=INT(RND*10)-5
340 IF Z=0 THEN 350 ELSE CAL
L SAY("GET+JOYSTICK+AND+PRES
S+IT+NUMBER+ONE")
350 CALL KEY(1,K,S):: IF S=0
 THEN 350 ELSE CALL SPRITE(#
25,140,2,145,226):: CALL MOT
ION(#25,0,-1):: CALL G
360 CALL SOUND(-2000,-7,10)
370 COL=INT(RND*13)+3 :: CAL
L SPRITE(#1,124,COL,ZE,SP)
380 CALL MOTION(#1,ZE1,SP1):
: CALL POSITION(#1,X1,Y1)::
IF X1<116 AND X1>32 AND Y1>3
2 AND Y1<216 THEN CALL COLOR
(#1,COL)ELSE CALL COLOR(#1,1
)
390 CALL JOYST(1,X,Y):: CALL
 KEY(1,K,S):: CALL MOTION(#1
,-Y*3,-(X*3)):: IF K=18 THEN
 410 ELSE CALL SOUND(-1000,-
7,10)
400 T=T-100 :: DISPLAY AT(23
,25)SIZE(-4):T :: IF T<=0 TH
EN 490 ELSE CALL POSITION(#2
5,X3,Y3):: IF Y3<112 THEN 54
0 ELSE 380
410 CALL POSITION(#1,X1,Y1):
: CALL SPRITE(#10,88,5,112,3
2,-70,127,#11,92,5,112,208,-
70,-127):: CALL SOUND(-150,-
5,0)
420 IF X1>57 AND X1<92 AND Y
```

```
1)102 AND Y1<140 THEN 450
430 CALL DELSPRITE(#10,#11):
: R(SI)=R(SI)-1 :: IF R(SI)<
=0 THEN 540 ELSE DISPLAY AT(
23,2)SIZE(2):USING "##":R(SI
)
440 GOTO 380
450 CALL DELSPRITE(#10,#11):
: CALL PATTERN(#1,136):: FOR
 I=0 TO 5 :: CALL SOUND(-150
,-7,I):: NEXT I
460 R(SI)=R(SI)-1 :: IF R(SI
)<=0 THEN 540 ELSE CALL DELS
PRITE(#1):: DISPLAY AT(23,2)
SIZE(2):USING "##":R(SI)
470 W=W+1 :: IF W>4 AND T>49
0 THEN CALL LOCATE(#25,145,2
26):: W=0
480 P(SI)=P(SI)+50 :: T=1000
 :: DISPLAY AT(23,25)SIZE(-4
):USING "#####":T :: GOTO 370
490 CALL MOTION(#1,0,0):: CA
LL SOUND(4250,-5,0):: FOR I=
1 TO 5
500 CALL COLOR(2,9,15,9,9,15
,10,9,15,11,9,15):: FOR J=1
TO 50 :: NEXT J
510 CALL COLOR(2,15,9,9,15,9
,10,15,9,11,15,9):: FOR J=1
TO 50 :: NEXT J :: NEXT I ::
 CALL COLOR(2,5,15,9,5,15,10
,5,15,11,5,15)
520 B(SI)=B(SI)-1 :: IF B(SI
)<5 THEN 540 ELSE CALL COLOR
(#B(SI),9):: T=1000 :: DISPL
AY AT(23,25)SIZE(4):T
530 GOTO 370
540 IF SI=2 THEN 590 ELSE CA
LL DELSPRITE(#1):: DISPLAY A
T(23,10)SIZE(-10):USING "###
########":NA$ :: T=1000 :: DI
SPLAY AT(23,25)SIZE(-4):T
550 CALL DELSPRITE(#25)
560 SI=SI+1 :: DISPLAY AT(23
,2)SIZE(-2):"40" :: CALL COL
OR(#5,13,#6,13,#7,13)
570 CALL G
580 IF Z=0 THEN 350 ELSE CAL
L SAY("GET+JOYSTICK+AND+PRES
S+IT+NUMBER+TWO+PLEASE"):: G
OTO 350
590 CALL G
600 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL B1 :: CALL
MAGNIFY(3)
610 CALL SOUND(-2000,-6,15)
620 SI=1 :: B(1),B(2)=2000
630 CALL SPRITE(#1,40,7,112,
32,#2,84,15,128,224,#3,44,2,
144,32,#4,44,2,144,224):: CA
LL SOUND(-2000,-6,15)
640 CALL JOYST(SI,X,Y):: CAL
L MOTION(#SI,-4*Y,4*X,#SI+2,
```

LISTINGS

```
0,4*X):: CALL KEY(SI,K,S)::
IF K=18 THEN 690
650 CALL COINC(#1,#2,10,W)::
 IF W THEN 830 ELSE CALL SOU
ND(-1000,-6,15)
660 CALL POSITION(#SI,X1,Y1)
:: IF X1<95 THEN CALL PATTER
N(#SI+2,88)ELSE CALL PATTERN
(#SI+2,44)
670 IF X1>152 OR X1<8 THEN 7
90
680 IF SI=1 THEN SI=2 :: GOT
O 640 ELSE SI=1 :: GOTO 640
690 IF SI=2 THEN 750 ELSE CA
LL MOTION(#1,0,0,#3,0,0):: C
ALL POSITION(#1,X1,Y1)
700 CALL SPRITE(#5,92,9,X1,Y
1,0,127):: CALL POSITION(#2,
X2,Y2):: IF ABS(X1-X2)<16 TH
EN 730
710 CALL DELSPRITE(#5):: B(1
)=B(1)-50 :: DISPLAY AT(23,7
)SIZE(4):USING "#####":B(1)::
 IF B(1)<=0 THEN 860 ELSE CA
LL SOUND(-2000,-6,15)
720 GOTO 640
730 CALL DELSPRITE(#4,#5)::
CALL PATTERN(#2,124):: FOR I
=0 TO 5 :: CALL SOUND(-150,-
7,I):: NEXT I :: CALL DELSPR
ITE(#2):: B(2)=B(2)-100
740 DISPLAY AT(23,23)SIZE(4)
:USING "#####":B(2):: IF B(2)
<=0 THEN 860 ELSE CALL SOUND
(-2000,-6,15):: GOTO 630
750 CALL MOTION(#2,0,0,#4,0,
0):: CALL POSITION(#2,X2,Y2)
760 CALL SPRITE(#5,92,9,X2,Y
2,0,-128):: CALL POSITION(#1
,X1,Y1):: IF ABS(X2-X1)<16 T
HEN 810
770 CALL DELSPRITE(#5):: B(2
)=B(2)-50 :: DISPLAY AT(23,2
3)SIZE(4):USING "#####":B(2):
: IF B(2)<=0 THEN 860 ELSE C
ALL SOUND(-2000,-6,15)
780 GOTO 640
790 CALL MOTION(#SI,0,0,#SI+
2,0,0):: CALL PATTERN(#SI,12
4):: FOR I=0 TO 5 :: CALL SO
UND(-150,-7,I):: NEXT I
800 B(SI)=B(SI)-100 :: IF B(
SI)<=0 THEN 860 ELSE IF SI=1
 THEN DISPLAY AT(23,7)SIZE(-
4):USING "#####":B(SI)ELSE DI
SPLAY AT(23,23)SIZE(-4):USIN
G "#####":B(SI):: GOTO 630
810 CALL DELSPRITE(#3,#5)::
CALL PATTERN(#1,124):: FOR I
=0 TO 5 :: CALL SOUND(-150,-
7,I):: NEXT I :: CALL DELSPR
ITE(#1):: B(1)=B(1)-100
820 DISPLAY AT(23,7)SIZE(4):
```

```
USING "####":B(1):: IF B(1)<
=Ø THEN 86Ø ELSE CALL SOUND(
-2ØØØ,-6,15):: GOTO 63Ø
83Ø CALL MOTION(#SI,Ø,Ø):: C
ALL DELSPRITE(#3,#4):: CALL
PATTERN(#1,124,#2,124):: FOR
 I=Ø TO 5 :: CALL SOUND(-15Ø
,-7,I):: NEXT I
84Ø CALL DELSPRITE(#1,#2)::
B(1)=B(1)-2ØØ :: B(2)=B(2)-2
ØØ :: IF B(1)<=Ø OR B(2)<=Ø
THEN 86Ø
85Ø CALL SOUND(-2ØØØ,-6,15):
: DISPLAY AT(23,7)SIZE(-4):U
SING "####":B(1):: DISPLAY A
T(23,23)SIZE(-4):USING "####
":B(2):: GOTO 63Ø
86Ø CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL CHARSET ::
A$(1)="ØØØØØØØ3Ø7ØC1818" ::
A$(2)="1818ØC07Ø3ØØØØØØ" ::
A$(3)="ØØØØØØCØEØ3Ø1818"
87Ø A$(4)="181830E0C0000000"
 :: A$(5)="1818181818181818"
 :: A$(6)="ØØØØØØFFFFØØØØØØ"
 :: FOR I=1 TO 6
88Ø CALL CHAR(34+I,A$(I))::
NEXT I :: FOR I=3 TO 11 :: C
ALL COLOR(I,13,2):: NEXT I :
: CALL COLOR(1,5,2,2,5,2)::
CALL SCREEN(2)
89Ø DISPLAY AT(1,5):"#((((((
((((((((((((%"¦TAB(5)¦"'SCOR
E-DISTRIBUTION'"¦TAB(5)¦"$((
((((((((((((((&"
9ØØ DISPLAY AT(6,2):"#((((((
((((((((((((((((%"¦TAB(2)¦
"'COMMANDER...STOOM...GOT.'"
¦TAB(2)¦"'.................
......'"¦TAB(2)¦"'.........
...POINTS.....'"
91Ø DISPLAY AT(1Ø,2):"$(((((
((((((((((((((((((&"
92Ø DISPLAY AT(13,2):"#(((((
((((((((((((((((((%"¦TAB(2)
¦"'COMMANDER...METU....GOT.'
"¦TAB(2)¦"'.................
.......'"¦TAB(2)¦"'.........
....POINTS.....'"
93Ø DISPLAY AT(17,2):"$(((((
((((((((((((((((((&"
94Ø DISPLAY AT(9,7)SIZE(4):U
SING "####":P(1):: DISPLAY A
T(16,7)SIZE(4):USING "####":
P(2)
95Ø DISPLAY AT(2Ø,9):"GO ON?
...Y" :: ACCEPT AT(2Ø,18)SIZ
E(-1)VALIDATE("YN")BEEP:AN$
96Ø IF AN$="Y" THEN 27Ø ELSE
 CALL CLEAR :: END
97Ø SUB B2
98Ø DEF SP=INT(RND*24)+5
99Ø DEF ZE=INT(RND*12)+5
```

LISTINGS

```
1ØØØ DATA 3C42818181814234C,,
,,ØØØØØØ1F1F1F1F1F,F8F8F8F8F
8ØØØØØØ,1F1F1F1F1FØØØØØØ,ØØØ
ØØØF8F8F8F8F8
1Ø1Ø DATA 3C7EFFFFFFFF7E3C,,
,,ØØ7E464A5666467E
1Ø2Ø DATA ØØ3Ø3Ø1Ø1Ø3C3C3C,Ø
Ø7EØ2Ø27E6Ø6Ø7E,ØØ7CØ4Ø47EØ6
Ø67E,ØØ4646467EØ2Ø2Ø2,ØØ7EØ4Ø
4Ø7EØ6Ø67E
1Ø3Ø DATA ØØ4Ø4Ø4Ø7E62627E,Ø
Ø7EØ2Ø6Ø6Ø6Ø2Ø2,ØØ7E42427E46
467E,ØØ7E46467EØ2Ø2Ø2,ØØØØ3Ø
3ØØØ3Ø3ØØØ
1Ø4Ø DATA ,,,,
1Ø5Ø DATA ØØ7E4Ø4Ø7E6Ø6Ø7E,Ø
Ø7E6Ø6Ø3E2Ø2Ø2Ø,ØØ4444447E66
6666,ØØØ8Ø8Ø818181818,ØØ2Ø2Ø
2Ø6Ø6Ø6Ø7E
1Ø6Ø DATA ØØ675B43435B5B5B,Ø
Ø7E42424646467E,ØØ7E42427E6Ø
6Ø6Ø,ØØ7E4247E Ø2627E,ØØ7EØ8
1818181Ø8Ø8
1Ø7Ø DATA ØØ4646464646467E,,
FF7F3F1FØF Ø7Ø3Ø1,FFFFFFFFFFF
FFFFF,
1Ø8Ø DATA ,Ø1Ø1Ø1Ø1Ø1Ø1Ø1Ø1,
8Ø8Ø8Ø8Ø8Ø8Ø8Ø8Ø8Ø,ØØØØØØØØØØ
ØØØFF,FFØØØØØØØØØØØØØ
1Ø9Ø DATA ,,,,
11ØØ DATA Ø1Ø2Ø4Ø81Ø2Ø4Ø8Ø,Ø
1Ø2Ø4Ø81Ø2Ø4Ø8Ø,,8Ø4Ø2Ø1ØØ8Ø
4Ø2Ø1,
111Ø DATA ,8Ø4Ø2Ø1ØØ8Ø4Ø2Ø1,
FFFEFCF8FØE Ø C Ø8Ø,FF7F3F1FØFØ
7Ø3Ø1,Ø1Ø3Ø7ØF1F3F7FFF
112Ø DATA 8ØCØEØFØF8FCFEFF,F
FFFFFFFFFFFFFFF,,FFFFFØF8DCC
EC7C3,C3C7CEDCF8FØFFFF
113Ø DATA FFFFØF1F3B73E3C3,C
3E3733B1FØFFFFF,C3C3C3C3C3C3
C3C3,FFFFØØØØØØØFFFF,Ø1Ø2Ø4
Ø81Ø2Ø4Ø8Ø
114Ø DATA 8Ø4Ø2Ø1ØØ8Ø4Ø2Ø1,Ø
ØØØØØF1F3F7FFF,ØØØØØFFFFFF
FFFF,FFFFFFFFFFØØØØØ,Ø1Ø3Ø7
ØF1FØØØØØ
115Ø DATA ØØØØØFØF8FCFEFF,8
ØCØEØFØF8ØØØØØ,F8F8F8F8F8F8
F8F8,1F1F1F1F1F1F1F1F,FFFFFF
FFFFFFFFFF
116Ø DATA ,ØØØØØØ1ØØØØØØØØ,
ØØØØØ1818ØØØØØØ,,
117Ø DATA Ø183874D391F3F75,D
FC663341EØC Ø7Ø3,8ØC1E1B29CF8
FCAE,FB63C62C783ØEØCØ,8ØCØEØ
FØF8FCFEFF
118Ø DATA meeeaddddddddddeed
ddddddddd'eeel
119Ø DATA emeeeaddddddddddeed
ddddddddd'eeele
12ØØ DATA eemeeeeeeeeeeeeeee
eeeeeeeeeelee
```

ABO SERVICE-KARTE

TI/Sp.

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

TI REVUE

Abo-Service Sp
Postfach 1107
8044 UNTERSCHLEISSHEIM

Coupon

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten zwölf Ausgaben an untenstehende Anschrift. Sollte ich nicht vier Wochen vor Ablauf schriftlich kündigen, läuft diese Abmachung automatisch weiter.

Name _____

Vorname _____

Straße/Hausnr. _____

Plz/Ort _____

Ich bezahle:

☐ per beiliegendem Verrechnungsscheck

☐ gegen Rechnung

☐ bargeldlos per Bankeinzug von meinem Konto

bei (Bank) und Ort _____

Kontonummer _____

Bankleitzahl _____

(steht auf jedem Kontoauszug)

Unterschrift _____

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift _____

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen

die Listings dieses Heftes auf

☐ Disketten zum Preis von DM 60,–

☐ Kassetten zum Preis von DM 60,–

Ich zahle:

Bar – per beigefügtem Geld ( )

per beigefügtem Scheck ( )

Gegen Bankabbuchung am Versandtag ( )

Zutreffendes bitte ankreuzen!

Meine Bank (mit Ortsname) .....

Meine Kontonummer .....

Meine Bankleitzahl ..... **(steht auf jedem Bankauszug)**

Vorname ..... **Nachname** .....

Str./Nr. ..... **PLZ / Ort** .....

Hiermit bestätige ich mit meiner Unterschrift, Ihre Verkaufsbedingungen gelesen zu haben und zu akzeptieren.

Unterschrift .....

Bitte ausschneiden und einsenden an
TI-REVUE
Special-Service 5/86
Postfach 1107
8044 Unterschleißheim

LESER WERBEN LESER

GEWINNEN SIE EINE COMPUTER-UHR! Und zusätzlich eventuell noch ein großes Computer-Buch. Oder ein Paket Disketten. ODER AUCH EINEN DRUCKER – ODER EINE DISKETTENSTATION! Wie? Sie werben einen Abonnenten. Dann haben Sie auf jeden Fall schon die Computer-Uhr gewonnen. Zusätzlich verlosen wir unter allen, die mitmachen, jeden Monat vier weitere wertvolle Preise. Und alle sechs Monate gibt es einen Hauptpreis unter allen Abo-Werbern zu gewinnen. Also: Mitmachen. Mitgewinnen.

Ja, ich mache mit beim Abo-Wettbewerb. Ich habe

Herrn/Frau _____

_____

Straße/Hausnr. _____

_____

Plz/Ort _____

_____

als neuen Abonnenten der TI-REVUE geworben.

Der neue Abonnent war bisher noch nicht Bezieher dieser Zeitschrift.
Als Prämie erhalte ich nach Eingang des Abo-Entgeltes auf jeden Fall eine Computer-Uhr, wie abgebildet, und nehme zusätzlich noch an der Verlosung des Monats sowie der halbjährlichen Hauptpreise teil. Mir ist bekannt, daß der Rechtsweg bei den Verlosungen ausgeschlossen ist.
Meinen Preis senden Sie an

Name _____

Straße/Hsnr. _____

Plz/Ort _____

(Bitte ausschneiden und zusammen mit der Abo-Bestellkarte links einsenden!)

ANZEIGENSERVICE

Die große Börse für jeden Zweck in der TI REVUE. Kostenlos für Privat-Inserenten. Spottbillig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Achtung! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt nicht TI-typisch ist oder gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatanbieter: maximal acht Zeilen à 28 Anschläge. Für gewerbliche Anbieter: 5 DM p. mm.

TI
Anzeigenabteilg.
Postfach 1107
8044 Lohhof

Name _____
Vorname _____
Straße/Hausnr. _____
PLZ/Ort _____

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE

TI REVUE
Das Magazin für TI 99-4A

KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre TI REVUE nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Ihr" Kiosk nicht beliefert wurde? Kein Problem! Für ganze 60 DM liefern wir per Post zwölf Hefte ins Haus (Ausland 80 DM). Einfach den Bestellschein auf der nächsten Seite ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). TI REVUE kommt dann pünktlich ins Haus.

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen– Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für zwölf Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

DAS ANGEBOT: KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen ab sofort die TI-REVUE: KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!

Wir veröffentlichen nur Kleinanzeigen privater Inserenten, keine gewerblichen Anzeigen. Die kosten pro Millimeter DM 5,– plus Mehrwertsteuer!

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!

Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!

Die Insertion ist nicht vom Kauf des Heftes abhängig!

RESERVIERUNGS-SERVICE

Selbstverständlich denken wir bei diesem Special-Heft auch wieder an jene unter unseren Lesern, die keine Zeit haben, diese ganzen Listings einzugeben. Für sie hält der Kassettenservice das gesamte Angebot dieses Heftes auf Disketten bereit. Diese Disketten sind nur im Paket zu beziehen, sie kosten geschlossen DM 60,–. Einfach den Coupon auf den nächsten Seiten ausschneiden, ausfüllen und absenden. Diese Seite ist so gestaltet, daß keine Information des Heftes verloren geht.

Natürlich berücksichtigen wir auch jene TI-User, die noch kein Disketten-Laufwerk besitzen. Für sie haben wir alle Listings dieses Heftes auf Kassetten zusammengefaßt. Das Programmpaket kostet komplett DM 60,–. Es kann nur geschlossen bezogen werden.

Benutzen Sie bitte auch den Bestellcoupon auf der nächsten Seite.

Ein wichtiger Hinweis: Wir versenden nicht per Nachnahme oder Vorausrechnung!

Sie können aber auch den Betrag auf eines unserer Konten überweisen. Hier die Kontonommern:
Postgiro – Amt München, Konto W. Seibt / TI-Verlag, Kto-Nr. 416 103-805, BLZ 700 100 80 oder
Raiffeisenbank Lohhof, Kto.-Nr. 306 959, BLZ 701 695 85.

Bitte Absender (komplett) deutlich schreiben – am besten in Druckbuchstaben! Und Hinweis, was gewünscht wird, nicht vergessen! Z.B. Disketten oder Kassetten TI-Special Nr. 5!

BÖRSE

Billig! Zu verk. Ex-Bas. m. Lehrgang a. Cass u. dt. Handbuch 170,–, Datenverw. u. Textprogr. 2 Module in dt. + 100,–, Drucker Interface 50,–, The Attack 15,–, Engl. Grammatik 5,– DM. Videospiele I a. Cass. Tel: 56800, Gelsenkirchen

---

ACHTUNG
Suche Software aller Art für TI-Ex.Bas + Cass. Schickt Eure Listen an: Holger Scherer, Jungstr. 25, 6550 Bad-Kreuznach

---

Verk. Epson-Drucker RX-80 zum Direktanschluß an den TI 99/4A m. Interface und Kabeln (ca. 3 J. alt). Nur 399,– DM!!! Schriftprobe und Info geg. Rückumschl. bei: B. Walter, Pfortengartenw. 57, 6230 Frankfurt/M 80

Verk. TI 99/4A, P-Box, Ram 32K, RS232, Disk-Drive, Ex-Bsic + 2 Plus, Mod. z. TV, Kass.Kabel Softw.: ID-Konto, Data, Copy, Div. Progr., Lern-Kass, Bücher: 99 Spez. 1 + 2, Ex-Basic, tech. Unterlagen 1000,– Fr. J. Bucher, Schützenmattring 5, CH-6374 Buochs, Tel: 041/643360

!Ges.*Wanted*Ges.!
TI-User i. Raum Nürnberg zum Programmtausch. Schickt Eure Liste an: L. Hambach, Regensburgerstr. 236, 8500 Nürnberg 30

Verk. TI99/4A Computer. Tel: 02152/510232, nach 18 Uhr

Verkaufe TI 99/4A + Rec.Kabel + H.-Buch + Netzteil + Kabelbox. Alles gut erhalten! Gegen Bargeld abholen. VB 150,– DM. Marco Schädlich, Hofstr. 22, 7119 Weißbach, Tel: 07947/2414

Verk. TI 99/4A + Ex. Basic + Peri Box + Disk-Laufw. + Contr. + 32 K-Erw. + E/Ass. + Diskmana + Literatur! VB 1400,– DM od. einzeln! Tel: 0711/2166880, v. 8 - 15 Uhr

---

Verkaufe TI-Module: Parsec 35,– (NP 50,–), Tombstone 20,– (NP 35,–), Buck Rogers 55,– (NP 80,–), Star Treck 50,– (NP 70,–), je. + Porto. NN o. Vorauskasse. Tel: 06245/6315

Sanyo-Monitor DM 2112 Grün 12" Bildschirm zu verk. Anschlußkabel f. C64/TI99/4A dabei, VB 150,– DM. Tel: 07664/4822, Wochenende

---

Verkaufe oder tausche orig. Disk-Controller PHP 1240 m. DM-Modul, neu! VB 280,– DM. Such P-Box, RS 232, 32K-RAM, Laufw., Speech-Synth., LOGO II P-Code, Modulexp. 3-fach. R. Buchholz, D. Bonhoefferstr. 10, 4156 Willich 1, Tel: 02154/2724

Verk. Thermal Printer Typ PHP orig. Texas, keine Schnittstelle notw., da direkt an Konsole anzuschl. Preis 200,– DM, Tel: 030/7865425

Suche: Speech Synthes.
Biete: Grafik Tablett Super-Scetch!!!
Melden bei: Christian Schäfer, Großburschlaerstr. 17, 3442 Wanfried 4

Achtung! Module für TI 99/4A Basiclehrgang, Alpiner, Munch Man, TI Invaders, Versandliste Datenverw., Market. Planspiel, Alligatormix, Early Reading, Progr.Routinen 1, Check-Book, Manag. Modul 30,– - 60,– DM. Tel: 040/6446058. Henrik Wedekind

---

Suche MBI-RS 232, Interface. Verk.: Orig. Netzteil m. Trafo 30,–, defekt. Konsole 25,–. Tel: 06181/73949

Biete: Ex.-B. + DT Ex-B. Handb. 250,–, MBI Interf. (RS 232) 95,–, leicht def. TI 99/4A + Zub. 70,–. CH Trail, Microsu., P. Position, M. Patrol, Finanzber., je 35,–, kompl. 500,– DM od. Tausch geg. „RS232" + „32 K" Karten. Jörg Klaas, Tel: 02562/1362

TI 99/4A zu verk.
TOP-Zustand! Zubehör: Ex-Bas., mit dt. Handbuch, Rec. Kabel, div. dt. Bücher, Programmhefe, 2 bespr. Kass. Tel: 06181/74669, Kompl. 450,– DM

*** Achtung***
Drucke Ihre Listing: 1 = 1 DM; 3 = 2 DM; 6 = 4 DM. Jedes weitere + 1 DM + 80 PF Porto. Knut Meissner, Marsbergerstr. 25, 3549 Diemelstadt 3

Basic für Anfänger (Cass.) für nur 6,– DM inkl. Porto. Tel: 06638/1503, ab 15 Uhr

---

TI 99/4A Konsole. Einwandfreier Zustand + Handbuch TI/Basic, Ex.-Basic, Preis VB. Harry Bauer, Helmtrudenstr. 11, 8000 München

---

Verk. folgende Module: Personal-Rec.-Keeping, Household-Bgt. Manag., je 40,–, (alle zus. 70,–); Statistik-Mod. 45,– (alle 3 Module 100,–); The Attack; Connect 4, Yathzee, Speech-Editor, je 20,– (alle zus. 60,–); Parsec 30,– DM. Tel: 07156/34941, D. Jarque

---

Miner 2049er. Das Spiel des Jahres 1983 – Ein Super-Modul für den TI 99/4A. Nur 45,– DM!!! Tel: 07156/34941

Verkaufe für den TI folgende Module: Pers. Rec. Keeping Household Bgt Mang., Statistics German, je 40,–, zus. 100,– DM.
Speech Editor, Connect Four, The Attack, Yathzee, je 20,–, zusm. 60,– DM.
Parsec 30,– DM + Porto + Verpack. Alle Module : 160,– DM. Tel: 07156/34941

Verk. TI-Writer 150,–, E/A 120,–, Statistik 50,–, Pers. Rep. 50,–, Text und Datei 50,–, Buchung 80,–, Fittnestraining 50,–, Hagera-Kurs 50,–, E/A Buch 50,–, Lager 30,–, Rechnung 30,–, Intern 20,–, Preise VB. Frank Brengel, Tel: 089/1231332

Verk. orig. neuw. intakte TI-Joyst. (1 Paar) 40,–, Rec. Kabel (ca. 1 m!) 50,–, zus. nur 50,– DM. NN od. Vorauskasse bei D. Augustin, Von Hasewinkelweg 87, 5020 Frechen 4

Verk. Grafik-Tablett. Super Sketch. VB 150,– DM. J. Günter, Freiheitstr. 3, 5802 Wetter 1, Tel: 02335/4190, ab 18 Uhr

---

Suche funktionsfähiges!!! Interface zwischen RS232 intern u. Silver Reed Ex 42 Typenradschreibmasch. Tel: 06084/2904

Verk. Disk.: Basicrout. II, Oig. TI 65,–, Basicrotu. III Orig. TI 65,–, Basivrout. H. A. Goerke 60,–, Diskmenü H. Martin 20,–, DAMAST Demodisk 30,–, Mathe I Orig. TI 20,–, Prog. + Spiele D. Karbach 50,–, Versandliste Orig. TI 65,–, Checkbookman. Orig. TI 65,–, Lagerverw. Orig. TI 65,–, Terminkal. A. Donix 30,–, Progr. TI-99 Journ. 55,– DM. Rudhard Thielmann, Tel: 02234/79646

An alle TI-Besitzer
Verk. Spitzenprogr. Informat. 1,– DM. Stephan Schiml, Wieselrieth 10, 8481 Leuchtenberg

---

Verk. TI 99/4A + 3 Module + Rec. Kabel + Handbüch. von TI, grundüberholt, alles 280,– DM, Wchenende Tel: 02151/399404 od. D. Schaertges, Bleibergerstr. 66, 5100 Aachen

---

Tausche Alpiner und Tombstone-City Modul geg. andere Module. Angebote an: Andreas Tilch, Weierstadt 17, 6648 Wadern-Büschfeld/Saar Tel: 06874/6705

Verkaufe oder tausche über 300 Progr. gegen Hardware. Angebote an/od. Listen anfordern bei: Manfred Schneider, Nunkircher Str. 42a, 6648 Wadern-Büschfeld/Saar, Tel: 06874/6705

Suche Bedienungsanleitung für Terminal Emulator II, auch Fotokopie, Tel: 07026/5885

Wer schenkt Jungen in DDR einen Comp.? Beliebiges System – funktionsfähig! Margot Neefe, Weißensteiner Str. 79, 7070 Schwäbisch Gmünd, Tel: 07171/85372

Disklaufw. orig. TI für Box 200,– DM od. Tausch geg. orig. Multiplan. TI-Joysticks 25,– DM. Behnisch, Tel: 02271/62511

---

32 K-Byte CMOS-Erw., m. echten 16Bit Datenbus, superschnell, zum Einbau in die Konsole, kompl. bestückt m. Anschlußkabel + Stecker + Einbauleitung (Löten erf.) Preis 120,– DM. Kann auch eingebaut werden. Tel: 06257/83247

TI-Peri.-Box m. orig. TI-Laufw. und Controller sowie DSK-Manager II. Modul, ca. 1 J. alt und 100 % o.k. für 750,– DM zu verk. Tel: 030/3911786. Grafik Table Sketch Master f. d. TI 99/4A zu verk. ca. 3 Mon. alt und orig. verp. f. 100,– DM. Tel: 030/391 1786

TI 99/4A 80,– DM: M-Mem 120,–, Schach, Datenverw. je 30,–, Parsec, Munchman etc. je 15,–, dt. engl. u. amerik. Literatur billig! J. Latza, Josefstr. 55, 4352 Herten, Tel: 02366/36806, nach 17.30 Uhr

---

LISTINGS

```
1210 DATA ceefkkkkkkkkkkkkkk
kkkkkkkkkkheeb
1220 DATA dcejMMMMMMMMMMMMMM
MMMMMMMMMMjebd
1230 DATA ddejMMMMMMMMMMMMMM
MMMMMMMMMMjedd
1240 DATA ddejMMMMMMMMMMMMMM
MMMMMMMMMMjedd
1250 DATA ddejMMMMMMMMMMMMMM
MMMMMMMMMMjedd
1260 DATA ddejMMMMMMMMMMMMMM
MMMMMMMMMMjedd
1270 DATA eeejMMMMMMMMMMMMMM
MMMMMMMMMMjeee
1280 DATA eeejMMMMMMMMMMMMMM
MMMMMMMMMMjeee
1290 DATA ddejMMMMMMMMMMMMMM
MMMMMMMMMMjedd
1300 DATA ddejMMMMMMMMMMMMMM
MMMMMMMMMMjedd
1310 DATA ddejMMMMMMMMMMMMMM
MMMMMMMMMMjedd
1320 DATA ddejMMMMMMMMMMMMMM
MMMMMMMMMMjedd
1330 DATA d'ejMMMMMMMMMMMMMM
MMMMMMMMMMjead
1340 DATA 'eegkkkkkkkkkkkkkk
kkkkkkkkkkieea
1350 DATA eelnooo+(oooooooooo
ooooooooormee
1360 DATA elbvvvvtuAJ@DKKKKK
KKKKKKKKKKLcme
1370 DATA lqppppp)*pppppppppp
pppppppppppsm
1380 DATA fkkkkkkhfkkkkkkkkk
kkkkkhfkkkkkkh
1390 DATA jHBFFIHjjGCDFIKKKK
KKKKKjjICE@KKj
1400 DATA jKK40KKjjKKHKIKFKF
KEKKKjjK1000Kj
1410 DATA gkkkkkkigkkkkkkkkk
kkkkkigkkkkkki
1420 RESTORE 1000 :: FOR I=3
6 TO 128 :: READ D$ :: CALL
CHAR(I,D$):: NEXT I
1430 CALL CLEAR :: CALL SCRE
EN(2):: FOR I=0 TO 14 :: CAL
L COLOR(I,2,2):: NEXT I
1440 RESTORE 1180 :: FOR I=1
 TO 24 :: READ D$ :: FOR J=1
 TO 32 :: CALL HCHAR(I,J,ASC
(SEG$(D$,J,1))):: NEXT J ::
NEXT I
1450 CALL HCHAR(19,29,128)
1460 CALL SCREEN(2):: CALL C
OLOR(2,5,15,3,2,13,4,2,13,5,
2,13,6,2,13)
1470 CALL COLOR(7,14,2,9,5,1
5,10,5,15,11,5,15,12,11,2,13
,13,5,1,2,13)
1480 FOR I=1 TO 25
1490 CALL HCHAR(ZE,SP,120)::
 NEXT I
1500 FOR I=1 TO 15
1510 CALL HCHAR(ZE,SP,121)::
 NEXT I
1520 FOR I=8 TO 9 :: CALL HC
HAR(I,16,80):: CALL HCHAR(I,
17,81):: CALL HCHAR(I+4,16,8
0):: CALL HCHAR(I+4,17,81)::
 NEXT I
1530 CALL HCHAR(10,14,82,2):
: CALL HCHAR(10,18,82,2):: C
ALL HCHAR(11,14,83,2):: CALL
 HCHAR(11,18,83,2)
1540 CALL SPRITE(#5,44,13,14
5,27,#6,44,13,145,37,#7,44,1
3,145,47)
1550 CALL CHAR(136,"01041044
0154002288015401441004010020
8822002A802289002A0022882000
")
1560 CALL CHAR(88,"000000000
00000000103060C183060C001030
60C183060C08000000000000000"
)
1570 CALL CHAR(92,"80C060301
80C06030100000000000000000000
0000000000080C06030180C0603"
)
1580 CALL CHAR(140,"00000018
18181818"&RPT$("0",48))
1590 SUBEND
1600 SUB B1
1610 DATA 3F3F3F7F7F7FFFFF,0
0000003071F3F7F,FFFFFFFFFFFF
FFFF,000103070 70F1F1F,011FFF
FFFFFFFFFF,80F8FFFFFFFFFFFF
1620 DATA 000000C0E0F8FCFE,0
080C0E0E0F0F8F8,FCFCFCFEFEFE
FEFF,FFFEFEFEFEFCFCFC,F8F8F0
E0E0C08,FEFCF8E0C
1630 DATA 3F0E3B073FFFF88,7F
3F1F0703,1F1F0F07070301,FF7F
7F7F7F3F3F3F,FC70DCE0FCFF1F0
1,FFF000000000FFFF
1640 DATA FF8FC7F183CFF1FF,9
FCFE9E3F1C7FFFF,FFFFFF8387F0
9EFC,C3F380FFFFF0FFFF,FFEFC7
87070F3F7,C33FE1870F1FFFFF
1650 RESTORE 1610 :: FOR I=9
6 TO 119 :: READ R$ :: CALL
CHAR(I,R$):: NEXT I
1660 CALL CHAR(58,"007E42427
E606060",59,"000808081818181
8",60,"002020206060607E",61,
"007E42424646467E")
1670 CALL CHAR(62,"007E08181
8180808",63,"007E60603E20202
0",64,"004646464646467E",65,
"007E40407E60607E")
1680 CALL CHAR(66,"003820202
0202038",67,"003808080808083
8",72,"010307173F3F7F7F",73,
"80E0E0F8FAFCFEFE")
1690 CALL CHAR(74,"3F9FCFCFC
```

```
7E7F1F8",75,"FCF9F3F3E3E78F1
F",76,"FFFFFFFFFFFFFFFF",80,
"FFFFFFFFFFFFFFFF")
1700 CALL CHAR(122,"",33,"FF
FFF0F8DCCEC7C3",35,"C3C3C3C3
C3C3C3C3",36,"C3E3733B1F0FFF
FF",37,"FFFF0F1F3B73E3C3")
1710 CALL CHAR(38,"C3C7CEDCF
8F0FFFF",39,"FFFF0000000000FFF
F",81,"FEFDFBF7EFDFBF7F",82,
"7FBFDFEFF7FBFDFE")
1720 CALL SCREEN(2):: FOR I=
0 TO 14 :: CALL COLOR(I,2,2)
:: NEXT I
1730 DATA zzzzzzzzzzzzzxzzzy
zzadbefzzzzyzx
1740 DATA zzxzzzxzzyzzzzzzzz
zcbbbbbgzxzzzz
1750 DATA zzzzzzzzzzzzxzzzzz
x'bbbbbhzzzzzz
1760 DATA xzzzxzzzxzzzzzzzzz
zbbbbbrszzzzxz
1770 DATA zzzzzzzzzzzzzzzzzz
zobbbtvizxzzzz
1780 DATA Izzzzzzzzzzzzyzzxz
znbrsuwjzzzHIz
1790 DATA LIyzxzHIzzzxzzzzzz
zzmpqlkzzyHLLI
1800 DATA LLIzzHLLIzzzzzzHIz
xzzzzzzzzHLLLL
1810 DATA LLLIHLLLLIHIzzHLLI
zzzzzxzzHLLLLL
1820 DATA LLLLJLLLLKLLIHLLLL
IxzHIzzHLLLLLL
1830 DATA LLLLLJLLKLLLKLLLLL
LIHLLIHLLLLLLL
1840 DATA LLLLLLJKLLLKJLLLLL
LLJLLKLLLLLLLL
1850 DATA PPPQPPPPPQPPPPPQRP
PPPPRPPPPPRPPP
1860 DATA PPPPPPPPPPPPPPPPP
PPPPPPPPPPPPP
1870 DATA PQPPPPPQPPPPPQPPPP
RPPPPPRPPPPPRP
1880 DATA QPPPPPQPPPPPQPPPPP
PRPPPPPRPPPPPR
1890 DATA PPPPPPPPPPPPPPPPP
PPPPPPPPPPPPP
1900 DATA PPPPPPPPPPPPPPPPP
PPPPPPPPPPPPP
1910 DATA PPPQPPPPPQPPPPPQRP
PPPPPRPPPPPRPPP
1920 DATA PPQPPPPPQPPPPPQPPR
PPPPPRPPPPPRPP
1930 DATA !''''''''''''''%!'
''''''''''''''%
1940 DATA #:;<=> 1.......##:
;<=> 2.......#
1950 DATA #?@A<..B2000C..##?
@A<..B2000C..#
1960 DATA &''''''''''''''$&'
''''''''''''''$
1970 RESTORE 1730 :: FOR I=1
 TO 24 :: READ R$ :: FOR J=1
 TO LEN(R$):: CALL HCHAR(I,J
,ASC(SEG$(R$,J,1))):: NEXT J
 :: NEXT I
1980 CALL SCREEN(7):: CALL C
OLOR(1,2,14,3,2,14,4,2,14,5,
2,14,6,5,2,7,3,2,9,11,2,10,1
1,2,11,11,2,12,12,2)
1990 CALL CHAR(40,"0000000004
0E0F17A7FFFEA7F0003000000000
00000E01008FCFEABFEC0F80000"
)
2000 CALL CHAR(84,"0000000000
007081037F7D57F031F000000000
0000207BF5EFEFF57FE00C00000"
)
2010 CALL CHAR(44,"0000000000
000000000000000001F3F7FFF00000
000000000000000000000F8FCFEFF"
)
2020 CALL CHAR(124,"0104104400
1540022880154014410040100200
8822002A802289002A0022882000
")
2030 CALL CHAR(92,"0000000F0
F000000000000F0F0000000000F0
F0F000000000000F0F")
2040 CALL CHAR(88,"0000000000
000000000000000F1F0000000000
000000000000000000000F0F80000"
)
2050 SUBEND
2060 SUB G
2070 FOR I=0 TO 9 :: CALL SO
UND(-250,391,I):: NEXT I ::
FOR I=0 TO 9 :: CALL SOUND(-
250,329,I,391,I+11):: NEXT I
 :: FOR I=0 TO 9
2080 CALL SOUND(-250,261,I,3
29,I+11,391,I+21):: NEXT I :
: SUBEND
3000 !Ich hoffe, Ihr habt da
     ran gedacht, dass je-
     der Punkt im Listing
     ein Leerzeichen dar-
     stellt !!!
```

Anmerkung der Redaktion:

Ganz besonders gut sind dem Autoren die Grafiken in den beiden Bildschirmen gelungen. Schon alleine deshalb war dieses Spiel veröffentlichungswert.

TI-REVUE
JEDEN MONAT
NEU

WOHNGELD

Wohngeld ist eine staatliche Hilfe als Zuschuß zur Bestreitung der Unterkunftskosten. Für Mieter heißt dieser Zuschuß Wohngeld, für Eigentümer eines Familienheimes oder einer Eigentumswohnung heißt es Lastenzuschuß. Im folgenden wird der Oberbegriff "Wohngeld" verwendet.
Viele Menschen wissen nicht, ob sie einen Anspruch auf die Gewährung von Wohngeld haben. Um dies zu ermitteln, bedarf es einiger "Rechnerei", denn ein etwaiger Anspruch auf Wohngeld hängt von verschiedenen Faktoren ab. Dazu gehören Familieneinkommen, Anzahl der zum Haushalt gehörenden Familienangehörigen, Datum der Bezugsfertigkeit der Wohnung, Höhe der Miete oder Belastung bis zu einem bestimmten Höchstbetrag, verschiedene Freibeträge etc.

Um einen etwaigen Anspruch auf Wohngeld sicher zu ermitteln, dazu dient dieses Programm.
Nach dem Programmdurchlauf wird angezeigt, ob ein solcher Anspruch besteht oder nicht. Die genaue Höhe des zu zahlenden Wohngeldes wird jedoch nicht angezeigt, da die Rechnerkapazität – auch mit der Speichererweiterung – hierfür nicht ausreicht. Aus diesem Grunde wird das nach dem Wohngeldgesetz zugrundezulegende Familieneinkommen und die anrechenbare Miete bzw. Belastung angezeigt. Mit diesen beiden Werten läßt sich die Höhe des Wohngeldes in einer Wohngeldtabelle, die bei jeder Gemeinde- oder Kreisverwaltung erhältlich ist, leicht ablesen.
Sollte ein Anspruch bestehen, so sollte man umgehend einen entsprechenden Wohngeldantrag stellen, denn Wohngeld wird nicht ab dem erstmaligen Bestehen eines Anspruchs, sondern erst ab Antragseingang bei der zuständigen Wohngeldbehörde gewährt.
Das Programm läuft im Dialogbetrieb ab, ist also benutzerfreundlich.
Nach dem Programmstart erfolgt eine kurze Programmeinweisung in Form von Erläuterungen.

Danach ist die Anzahl der zum Familienhaushalt gehörenden Personen anzugeben, außerdem das Jahr der erstmaligen Bezugsfertigkeit der Wohnung (4stellig). Anschließend wird gefragt, ob man Mieter einer Mietwohnung oder Eigentümer einer Eigentumswohnung oder eines Eigenheims ist.

a) Mieter
Es ist zunächst die Höhe der Bruttomiete einzugeben. Außerdem wird gefragt, ob in dieser Miete bereits Kosten für Garage, Wasserverbrauch, Strom oder Müllabfuhr enthalten sind, oder ob diese Kosten separat anfallen.

b) Eigentümer
Hier wird zunächst nach den monatlichen Zins- und Tilgungsleistungen gefragt. Daneben ist zur Ermittlung der berücksichtigungsfähigen Bewirtschaftungskosten die Wohnfläche anzugeben. Außerdem ist einzugeben, ob zu der Wohnung eine Garage gehört und ob ein Aufwendungsdarlehen bzw. -zuschuß aus öffentlichen Mitteln gewährt wird.
Zu beachten ist, daß alle diese Angaben sich nur auf den von der Familie selbstgenutzten Wohnraum beziehen.
Von hier ab geht die Berechnung für Mieter und Eigentümer wieder gemeinsam weiter.
Es wird nun nach der Anzahl der Einkommensbezieher unter den Haushaltsangehörigen gefragt. (Je nach der hier eingegebenen Zahl werden entsprechend viele Schleifen durchlaufen.)
Unter Einkommen ist Erwerbseinkommen, Nebeneinkünfte, Ausbildungsvergütungen, Renten, Unterhaltszahlungen etc. zu verstehen, Kindergeldleistungen bleiben außer Betracht.
Nach der Eingabe der Einkünfte der ersten Person sind deren Werbungskosten anzugeben. Diese entnimmt man dem letztjährigen Einkommen- oder Lohnsteuerjahresausgleich. Dann wird gefragt, ob die Person zu mindestens 80% schwerbehindert und/oder Kind zwischen 16 und 25 Jahren mit eigenen Einkünften ist.
Anschließend ist einzugeben, ob die Person Steuern, Renten- und Krankenversicherung bezahlt oder Teile davon. Als Rentenversicherung gilt bei Beamten auch eine Lebensversicherung (Kapitalversicherung), für die Monatsbeiträge von mindestens 30,– DM zu entrichten sind.
Als Krankenversicherung zählt auch eine private Krankenversicherung. Nach diesen Eingaben wird nach denselben Daten der zweiten Person gefragt, danach die der dritten etc., bis die Zahl der Einkommensbezieher abgearbeitet ist.
Danach erfolgt die Anzeige, ob ein Anspruch auf Wohngeld besteht oder nicht. Außerdem wird das Familieneinkommen und die höchstzuschußfähige Miete/Belastung angezeigt, damit anhand dieser Werte der genaue DM-Betrag in einer Wohngeldtabelle abgelesen werden kann.
Das Programm ist für die Mietenstufe 3 ausgelegt. Dies ist für die weitaus größte Zahl der Gemeinden in Deutschland zutreffend, da bei einem Anspruch in der Gemeindengrößenklasse 3 auch ein Anspruch – sogar noch höher – in den Stufen 4 und 5 besteht.
Wohnt man jedoch in einer Gemeinde der Stufe 1 oder 2, müßte das Programm geringfügig modifiziert werden. Interessierte User können sich hierzu jederzeit an mich wenden.

Thomas Ehlert

LISTINGS

```
10 !************************
11 !*                      *
12 !*       WOHNGELD       *
13 !*                      *
14 !*                      *
15 !*  Copyright T.Ehlert  *
16 !*                      *
17 !*      dipl.-vwwt      *
18 !*                      *
19 !*     Renoisstr. 25    *
20 !*                      *
21 !*     5300  Bonn  1    *
22 !*                      *
23 !* Benoetigte Geraete : *
24 !*                      *
25 !* TI 99/4A + Ext. Bas. *
26 !*                      *
27 !*   Speicherbelegung:  *
28 !*                      *
29 !*      13713 Bytes     *
30 !*                      *
31 !*                      *
32 !************************
33 CALL SCREEN(16)
34 CALL CLEAR
35 DISPLAY AT(5,1):".....W
O H N G E L D.............
............."
36 FOR I=7 TO 21 :: CALL SOU
ND(75,160*I/5,0):: DISPLAY A
T(7,I):"*" :: NEXT I
37 CALL SOUND(500,160,0,240,
0,320,0)
38 DISPLAY AT(10,1):"Program
m zur Feststellung...eines e
twaigen Anspruchs aufWohngel
d oder Lastenzuschussnach de
m Wohngeldgesetz 1986"
39 DISPLAY AT(16,1):".......
@..T. Ehlert"
40 DISPLAY AT(18,1):".......
.dipl.-vwwt"
41 DISPLAY AT(20,1):".....Te
l.:0228/215991"
42 DISPLAY AT(23,10):"Bitte
Taste druecken"
43 CALL KEY(0,K,S):: IF S=0
THEN 43
44 CALL CLEAR
45 DISPLAY AT(1,1):"Im folge
nden wird Ihr Fami- lieneink
ommen und die fuer..Ihre Woh
nung hoechstzuschussfaehige
Miete/Belastung er- mittelt"
46 DISPLAY AT(6,1):"Nachdem
Sie die Anzahl der..zu Ihrer
 Familie gehoerendenPersonen
 eingegeben haben,..muessen
Sie noch das Jahr...der Bezu
gsfertigkeit Ihrer"
47 DISPLAY AT(11,1):"Wohnung
 angeben (4-stellig)"
48 DISPLAY AT(12,1):"Danach
muessen Sie angeben, ob Sie
Mieter oder Eigentue-mer Ihr
er jetzigen Wohnung  sind."
49 DISPLAY AT(17,1):"Im Ansc
hluss daran muessen..Sie die
 Komponenten Ihrer...Miete b
zw. Belastung angebenWelche
Bestandteile dazu ge-hoeren,
 zeigt das Programm."
50 DISPLAY AT(23,10):"Bitte
Taste druecken"
51 CALL KEY(0,K,S):: IF S=0
THEN 51
52 CALL CLEAR
53 DISPLAY AT(1,1):"Auf glei
che Weise fuehrt Siedas Prog
ramm durch die ein- zelnen S
chritte der Einkom- mensermi
ttlung."
54 DISPLAY AT(5,1):"Zum Schl
uss wird angezeigt, ob Sie e
inen Anspruch auf...Gewaehru
ng von Wohngeld ha- ben. Ist
 dies der Fall, so"
55 DISPLAY AT(9,1):"sollten
Sie schnellstens beiIhrer Ge
meinde- oder Kreis- verwaltu
ng einen entspre-...chenden
Antrag stellen, denn"
56 DISPLAY AT(13,1):"Wohngel
d wird nicht rueck-..wirkend
, sondern erst ab An-tragsei
ngang gewaehrt."
57 DISPLAY AT(16,1):"Neben d
em grundsaetzlichen  Anspruc
h wird noch Ihr Fami-lienein
kommen und Ihre anre-chenbar
e Miete/Belastung an-"
58 DISPLAY AT(20,1):"gezeigt
. Damit und mit einerWohngel
dtabelle laesst sich Ihr Ans
pruch in DM ablesen." :: DIS
PLAY AT(23,10):"Bitte Taste
dreucken"
59 CALL KEY(0,K,S):: IF S=0
THEN 59
60 !
61 !
62 ! Berechnung hoechstzusch
ussfaehige Miete
63 !
64 !
65 CALL SCREEN(12)
66 CALL CLEAR
67 DISPLAY AT(3,1):"Zahl der
 Fam.Angehoerigen? "
68 ACCEPT AT(3,28)VALIDATE("
12345")SIZE(1)BEEP:F
69 DISPLAY AT(5,1):"Jahr d.B
ezugsfertigkeit?"
70 ACCEPT AT(5,25)VALIDATE(N
UMERIC)SIZE(4)BEEP:J
71 IF J<1966 AND F=1 THEN MM
```

```
AX=320
72 IF J<1966 AND F=2 THEN MM
AX=415
73 IF J<1966 AND F=3 THEN MM
AX=495
74 IF J<1966 AND F=4 THEN MM
AX=575
75 IF J<1966 AND F=5 THEN MM
AX=660
76 IF J>1965 AND J<1972 AND
F=1 THEN MMAX=360
77 IF J>1965 AND J<1972 AND
F=2 THEN MMAX=465
78 IF J>1965 AND J<1972 AND
F=3 THEN MMAX=555
79 IF J>1965 AND J<1972 AND
F=4 THEN MMAX=645
80 IF J>1965 AND J<1972 AND
F=5 THEN MMAX=735
81 IF J>1971 AND J<1978 AND
F=1 THEN MMAX=380
82 IF J>1971 AND J<1978 AND
F=2 THEN MMAX=490
83 IF J>1971 AND J<1978 AND
F=3 THEN MMAX=585
84 IF J>1971 AND J<1978 AND
F=4 THEN MMAX=680
85 IF J>1971 AND J<1978 AND
F=5 THEN MMAX=775
86 IF J>=1978 AND F=1 THEN M
MAX=405
87 IF J>=1978 AND F=2 THEN M
MAX=525
88 IF J>=1978 AND F=3 THEN M
MAX=625
89 IF J>=1978 AND F=4 THEN M
MAX=730
90 IF J>=1978 AND F=5 THEN M
MAX=830
91 CALL CLEAR
92 DISPLAY AT(2,1):"Es ist j
etzt der Hoechstbe- trag der
 zuschussfaehigen  Miete od
er Belastung zu er- mitteln.
"
93 DISPLAY AT(7,1):"Zur Miet
e gehoeren auch die Kosten d
es Wasserverbrauchs,der Abwa
sser- und Muellbe-  seitigun
g, aber nicht Gara- gen- Str
om- und Heizkosten."
94 DISPLAY AT(13,1):"Unter B
elastung versteht mandie Auf
wendungen fuer den...Kapital
dienst und die Bewirtschaftu
ng eines Eigenheims  oder ei
ner Eigentumswohnung"
95 DISPLAY AT(19,1):"Sind Si
e..Mieter.....(M)...oder...
...Eigentuemer (E)?.........
.....Bitte B
uchstaben eingeben"
96 ACCEPT AT(22,28)VALIDATE(
```

LISTINGS

```
"ME")SIZE(1)BEEP:S$
97 IF S$="M" THEN 104
98 IF S$="E" THEN 130
99 !
100 !
101 ! Berechnung fuer Mieter
102 !
103 !
104 CALL CLEAR
105 DISPLAY AT(2,7):"MIETBES
TANDTEILE"
106 DISPLAY AT(5,2):"Bruttom
iete:"
107 ACCEPT AT(5,17)VALIDATE(
NUMERIC)SIZE(-4)BEEP:BRM
108 DISPLAY AT(7,2):"In dies
er Bruttomiete sind  folgend
e Nebenkosten bereitsenthalt
en:"
109 DISPLAY AT(12,5):"Garage
:"
110 ACCEPT AT(12,18)VALIDATE
(NUMERIC)SIZE(2)BEEP:GA
111 DISPLAY AT(14,5):"Heizko
sten:"
112 ACCEPT AT(14,18)VALIDATE
(NUMERIC)SIZE(3)BEEP:HK
113 DISPLAY AT(16,5):"Stromk
osten:"
114 ACCEPT AT(16,18)VALIDATE
(NUMERIC)SIZE(3)BEEP:SK
115 DISPLAY AT(18,2):"Zusaet
zlich zur Bruttomiete fallen
 an:"
116 DISPLAY AT(22,5):"Wasser
geld:"
117 ACCEPT AT(22,18)VALIDATE
(NUMERIC)SIZE(3)BEEP:WG
118 DISPLAY AT(24,5):"Muella
bfuhr:"
119 ACCEPT AT(24,18)VALIDATE
(NUMERIC)SIZE(3)BEEP:MA
120 KM=BRM-GA-HK-SK+WG+MA
121 IF KM=MMAX THEN M=KM
122 IF KM>MMAX THEN M=MMAX
123 IF KM<MMAX THEN M=KM
124 GOTO 155
125 !
126 !
127 ! Berechnung fuer Eigent
uemer
128 !
129 !
130 CALL CLEAR
131 DISPLAY AT(2,1):"BESTAND
TEILE DER BELASTUNG:"
132 DISPLAY AT(3,10):"(monat
lich)"
133 DISPLAY AT(5,5):"Zinsen:
"
134 ACCEPT AT(5,18)VALIDATE(
NUMERIC)SIZE(-7)BEEP:ZI
135 DISPLAY AT(7,5):"Tilgung
```

```
:"
136 ACCEPT AT(7,18)VALIDATE(
NUMERIC)SIZE(-7)BEEP:TI
137 DISPLAY AT(10,2):"BEWIRT
SCHAFTUNGSKOSTEN"
138 DISPLAY AT(13,5):"Wohnfl
aeche in qm:"
139 ACCEPT AT(13,26)VALIDATE
(NUMERIC)SIZE(-3)BEEP:WFL
140 BWK=(WFL*20)/12
141 DISPLAY AT(15,5):"Garage
? (J-N)"
142 ACCEPT AT(15,26)VALIDATE
("JN")SIZE(1)BEEP:G$
143 IF G$="J" THEN GG=40
144 DISPLAY AT(17,5):"Aufwen
dungsdarlehen:"
145 ACCEPT AT(17,26)VALIDATE
(NUMERIC)SIZE(-3)BEEP:AWD
146 BEL=ZI+TI+BWK-GG-AWD
147 IF BEL=MMAX THEN M=BEL
148 IF BEL<MMAX THEN M=BEL
149 IF BEL>MMAX THEN M=MMAX
150 !
151 !
152 ! Einkommensermittlung
153 !
154 !
155 CALL CLEAR
156 DISPLAY AT(2,1):"Nachdem
 nun die hoechstzu- schussf
aehige Miete bzw. Be-lastung
 errechnet ist, muss das anr
echenbare Familien-"
157 DISPLAY AT(6,1):"einkomm
en aller zum Haushaltgehoere
nden Personen ermit- telt we
rden. Hierzu geben...Sie bit
te nun das Bruttojah-"
158 DISPLAY AT(10,1):"resein
kommen jeder einzelnenPerson
 hintereinander ein,..so wie
 auf dem Bildschirm...angeze
igt."
159 DISPLAY AT(16,1):"Zum Br
uttojahreseinkommen...zaehle
n Loehne, Gehaelter,..Arbeit
slosengeld, Kranken-..geld,
Renten, Unterhaltszah-lungen
, aber KEIN Kindergeld"
160 DISPLAY AT(22,1):"Wievie
le Familienangehoerigehaben
Einkommen"
161 ACCEPT AT(23,28)VALIDATE
(NUMERIC)SIZE(1)BEEP:AN
162 CALL KEY(0,K,S):: IF S=0
 THEN 162
163 FOR I=1 TO AN
164 CALL CLEAR
165 DISPLAY AT(1,3):"EINKOMM
EN VON PERSON";I;
166 DISPLAY AT(3,1):"Bruttoj
ahreseinkommen:"
167 ACCEPT AT(3,24)VALIDATE(
NUMERIC)SIZE(-5)BEEP:BJ(I)
168 DISPLAY AT(5,1):"Werbung
skosten:"
169 ACCEPT AT(5,24)VALIDATE(
NUMERIC)SIZE(-5)BEEP:WBK(I)
170 IF WBK(I)<564 THEN WBK(I
)=564
171 DISPLAY AT(7,1):"80% sch
werbehindert (J/N)?" :: ACCE
PT AT(7,28)VALIDATE("JN")SIZ
E(1)BEEP:SW$
172 IF SW$="J" THEN SBF(I)=2
400
173 DISPLAY AT(9,1):"Kind vo
n 16-25 Jahren(J/N)?" :: ACC
EPT AT(9,28)VALIDATE("JN")SI
ZE(1)BEEP:KI$
174 IF KI$="J" THEN KFB(I)=1
200
175 JEK(I)=BJ(I)-WBK(I)-SBF(
I)-KFB(I)
176 !
177 !
178 ! Pauschalfreibetraege
179 !
180 !
181 X=6
182 DISPLAY AT(12,1):"ZAHLEN
 SIE"
183 DISPLAY AT(14,1):"Steuer
n.........? (J=1/N=2)" :: ACC
EPT AT(14,28)VALIDATE("12")S
IZE(1)BEEP:ST
184 DISPLAY AT(16,1):"Kranke
nversich.? (J=1/N=2)" :: ACC
EPT AT(16,28)VALIDATE("12")S
IZE(1)BEEP:KV
185 DISPLAY AT(18,1):"Renten
versich. ? (J=1/N=2)" :: ACC
EPT AT(18,28)VALIDATE("12")S
IZE(1)BEEP:RV
186 IF ST=1 AND KV=1 AND RV=
1 THEN X=30 :: GOTO 189
187 IF ST=1 AND KV=1 OR ST=1
 AND RV=1 OR KV=1 AND RV=1 T
HEN X=20 :: GOTO 189
188 IF ST=1 OR KV=1 OR RV=1
THEN X=12.5 :: GOTO 189
189 PFB(I)=((JEK(I)/100)*X)
190 PEK(I)=JEK(I)-PFB(I)
191 NEXT I
192 FEK=PEK(1)+PEK(2)+PEK(3)
+PEK(4)+PEK(5)
193 E=FEK/12
194 ON F GOTO 200,229,264,30
3,346
195 !
196 !
197 !Einpersonenhaushalt
198 !
199 !
200 IF E<240 AND M>29.99 THE
```

```
N 389 ELSE 201
201 IF E<320 AND M>40 THEN 3
89 ELSE 202
202 IF E<380 AND M>50 THEN 3
89 ELSE 203
203 IF E<420 AND M>60 THEN 3
89 ELSE 204
204 IF E<460 AND M>70 THEN 3
89 ELSE 205
205 IF E<500 AND M>80 THEN 3
89 ELSE 206
206 IF E<560 AND M>90 THEN 3
89 ELSE 207
207 IF E<620 AND M>100 THEN
389 ELSE 208
208 IF E<700 AND M>120 THEN
389 ELSE 209
209 IF E<760 AND M>140 THEN
389 ELSE 210
210 IF E<820 AND M>160 THEN
389 ELSE 211
211 IF E<860 AND M>180 THEN
389 ELSE 212
212 IF E<920 AND M>200 THEN
389 ELSE 213
213 IF E<960 AND M>220 THEN
389 ELSE 214
214 IF E<1000 AND M>240 THEN
 389 ELSE 215
215 IF E<1020 AND M>260 THEN
 389 ELSE 216
216 IF E<1060 AND M>280 THEN
 389 ELSE 217
217 IF E<1080 AND M>300 THEN
 389 ELSE 218
218 IF E<1120 AND M>320 THEN
 389 ELSE 219
219 IF E<1140 AND M>340 THEN
 389 ELSE 220
220 IF E<1160 AND M>360 THEN
 389 ELSE 221
221 IF E<1180 AND M>380 THEN
 389 ELSE 222
222 IF E<1200 AND M>400 THEN
 389 ELSE 398
223 STOP
224 !
225 !
226 ! Zweipersonenhaushalt
227 !
228 !
229 IF E<340 AND M>29.99 THE
N 389 ELSE 230
230 IF E<400 AND M>40 THEN 3
89 ELSE 231
231 IF E<460 AND M>50 THEN 3
89 ELSE 232
232 IF E<520 AND M>60 THEN 3
89 ELSE 233
233 IF E<560 AND M>70 THEN 3
89 ELSE 234
234 IF E<600 AND M>80 THEN 3
89 ELSE 235
```

LISTINGS

```
235 IF E<640 AND M>90 THEN 3
89 ELSE 236
236 IF E<720 AND M>100 THEN
389 ELSE 237
237 IF E<820 AND M>120 THEN
389 ELSE 238
238 IF E<900 AND M>140 THEN
389 ELSE 239
239 IF E<980 AND M>160 THEN
389 ELSE 240
240 IF E<1040 AND M>180 THEN
 389 ELSE 241
241 IF E<1100 AND M>200 THEN
 389 ELSE 242
242 IF E<1160 AND M>220 THEN
 389 ELSE 243
243 IF E<1220 AND M>240 THEN
 389 ELSE 244
244 IF E<1280 AND M>260 THEN
 389 ELSE 245
245 IF E<1320 AND M>280 THEN
 389 ELSE 246
246 IF E<1360 AND M>300 THEN
 389 ELSE 247
247 IF E<1400 AND M>320 THEN
 389 ELSE 248
248 IF E<1440 AND M>340 THEN
 389 ELSE 249
249 IF E<1480 AND M>360 THEN
 389 ELSE 250
250 IF E<1500 AND M>380 THEN
 389 ELSE 251
251 IF E<1540 AND M>400 THEN
 389 ELSE 252
252 IF E<1580 AND M>420 THEN
 389 ELSE 253
253 IF E<1600 AND M>440 THEN
 389 ELSE 254
254 IF E<1620 AND M>460 THEN
 389 ELSE 255
255 IF E<1660 AND M>480 THEN
 389 ELSE 256
256 IF E<1680 AND M>500 THEN
 389 ELSE 257
257 IF E<1700 AND M>520 THEN
 389 ELSE 398
258 STOP
259 !
260 !
261 ! Dreipersonenhaushalt
262 !
263 !
264 IF E<420 AND M>39.99 THE
N 389 ELSE 265
265 IF E<480 AND M>50 THEN 3
89 ELSE 266
266 IF E<540 AND M>60 THEN 3
89 ELSE 267
267 IF E<580 AND M>70 THEN 3
89 ELSE 268
268 IF E<640 AND M>80 THEN 3
89 ELSE 269
269 IF E<700 AND M>90 THEN 3
```

```
89 ELSE 270
270 IF E<800 AND M>100 THEN
389 ELSE 271
271 IF E<900 AND M>120 THEN
389 ELSE 272
272 IF E<1000 AND M>140 THEN
 389 ELSE 273
273 IF E<1100 AND M>160 THEN
 389 ELSE 274
274 IF E<1200 AND M>180 THEN
 389 ELSE 275
275 IF E<1280 AND M>200 THEN
 389 ELSE 276
276 IF E<1340 AND M>220 THEN
 389 ELSE 277
277 IF E<1420 AND M>240 THEN
 389 ELSE 278
278 IF E<1480 AND M>260 THEN
 389 ELSE 279
279 IF E<1540 AND M>280 THEN
 389 ELSE 280
280 IF E<1580 AND M>300 THEN
 389 ELSE 281
281 IF E<1640 AND M>320 THEN
 389 ELSE 282
282 IF E<1680 AND M>340 THEN
 389 ELSE 283
283 IF E<1720 AND M>360 THEN
 389 ELSE 284
284 IF E<1760 AND M>380 THEN
 389 ELSE 285
285 IF E<1800 AND M>400 THEN
 389 ELSE 286
286 IF E<1840 AND M>420 THEN
 389 ELSE 287
287 IF E<1880 AND M>440 THEN
 389 ELSE 288
288 IF E<1900 AND M>460 THEN
 389 ELSE 289
289 IF E<1940 AND M>480 THEN
 389 ELSE 290
290 IF E<1960 AND M>500 THEN
 389 ELSE 291
291 IF E<2000 AND M>520 THEN
 389 ELSE 292
292 IF E<2020 AND M>540 THEN
 389 ELSE 293
293 IF E<2040 AND M>560 THEN
 389 ELSE 294
294 IF E<2080 AND M>580 THEN
 389 ELSE 295
295 IF E<2100 AND M>600 THEN
 389 ELSE 296
296 IF E<2120 AND M>620 THEN
 389 ELSE 398
297 STOP
298 !
299 !
300 ! Vierpersonenhaushalt
301 !
302 !
303 IF E<540 AND M>49.99 THE
N 389 ELSE 304
```

LISTINGS

```
304 IF E<600 AND M>60 THEN 3
89 ELSE 305
305 IF E<680 AND M>70 THEN 3
89 ELSE 306
306 IF E<760 AND M>80 THEN 3
89 ELSE 307
307 IF E<860 AND M>90 THEN 3
89 ELSE 308
308 IF E<960 AND M>100 THEN
389 ELSE 309
309 IF E<1100 AND M>120 THEN
 389 ELSE 310
310 IF E<1240 AND M>140 THEN
 389 ELSE 311
311 IF E<1340 AND M>160 THEN
 389 ELSE 312
312 IF E<1460 AND M>180 THEN
 389 ELSE 313
313 IF E<1560 AND M>200 THEN
 389 ELSE 314
314 IF E<1660 AND M>220 THEN
 389 ELSE 315
315 IF E<1740 AND M>240 THEN
 389 ELSE 316
316 IF E<1820 AND M>260 THEN
 389 ELSE 317
317 IF E<1880 AND M>280 THEN
 389 ELSE 318
318 IF E<1960 AND M>300 THEN
 389 ELSE 319
319 IF E<2020 AND M>320 THEN
 389 ELSE 320
320 IF E<2080 AND M>340 THEN
 389 ELSE 321
321 IF E<2120 AND M>360 THEN
 389 ELSE 322
322 IF E<2180 AND M>380 THEN
 389 ELSE 323
323 IF E<2240 AND M>400 THEN
 389 ELSE 324
324 IF E<2280 AND M>420 THEN
 389 ELSE 325
325 IF E<2320 AND M>440 THEN
 389 ELSE 326
326 IF E<2360 AND M>460 THEN
 389 ELSE 327
327 IF E<2400 AND M>480 THEN
 389 ELSE 328
328 IF E<2440 AND M>500 THEN
 389 ELSE 329
329 IF E<2480 AND M>520 THEN
 389 ELSE 330
330 IF E<2520 AND M>540 THEN
 389 ELSE 331
331 IF E<2540 AND M>560 THEN
 389 ELSE 332
332 IF E<2580 AND M>580 THEN
 389 ELSE 333
333 IF E<2620 AND M>600 THEN
 389 ELSE 334
334 IF E<2640 AND M>620 THEN
 389 ELSE 335
335 IF E<2660 AND M>640 THEN
```

```
 389 ELSE 336
336 IF E<2700 AND M>660 THEN
 389 ELSE 337
337 IF E<2720 AND M>680 THEN
 389 ELSE 338
338 IF E<2740 AND M>700 THEN
 389 ELSE 339
339 IF E<2780 AND M>720 THEN
 389 ELSE 398
340 STOP
341 !
342 !
343 ! Fuenfpersonenhaushalt
344 !
345 !
346 IF E<560 AND M>49.99 THE
N 389 ELSE 347
347 IF E<640 AND M>60 THEN 3
89 ELSE 348
348 IF E<720 AND M>70 THEN 3
89 ELSE 349
349 IF E<820 AND M>80 THEN 3
89 ELSE 350
350 IF E<920 AND M>90 THEN 3
89 ELSE 351
351 IF E<1040 AND M>100 THEN
 389 ELSE 352
352 IF E<1200 AND M>120 THEN
 389 ELSE 353
353 IF E<1340 AND M>140 THEN
 389 ELSE 354
354 IF E<1460 AND M>160 THEN
 389 ELSE 355
355 IF E<1580 AND M>180 THEN
 389 ELSE 356
356 IF E<1680 AND M>200 THEN
 389 ELSE 357
357 IF E<1780 AND M>220 THEN
 389 ELSE 358
358 IF E<1860 AND M>240 THEN
 389 ELSE 359
359 IF E<1960 AND M>260 THEN
 389 ELSE 360
360 IF E<2020 AND M>280 THEN
 389 ELSE 361
361 IF E<2100 AND M>300 THEN
 389 ELSE 362
362 IF E<2180 AND M>320 THEN
 389 ELSE 363
363 IF E<2240 AND M>340 THEN
 389 ELSE 364
364 IF E<2300 AND M>360 THEN
 389 ELSE 365
365 IF E<2360 AND M>380 THEN
 389 ELSE 366
366 IF E<2420 AND M>400 THEN
 389 ELSE 367
367 IF E<2460 AND M>420 THEN
 389 ELSE 368
368 IF E<2520 AND M>440 THEN
 389 ELSE 369
369 IF E<2560 AND M>460 THEN
 389 ELSE 370
370 IF E<2620 AND M>480 THEN
 389 ELSE 371
371 IF E<2660 AND M>500 THEN
 389 ELSE 372
372 IF E<2700 AND M>520 THEN
 389 ELSE 373
373 IF E<2740 AND M>540 THEN
 389 ELSE 374
374 IF E<2780 AND M>560 THEN
 389 ELSE 375
375 IF E<2800 AND M>580 THEN
 389 ELSE 376
376 IF E<2840 AND M>600 THEN
 389 ELSE 377
377 IF E<2880 AND M>620 THEN
 389 ELSE 378
378 IF E<2920 AND M>640 THEN
 389 ELSE 379
379 IF E<2940 AND M>660 THEN
 389 ELSE 380
380 IF E<2980 AND M>680 THEN
 389 ELSE 381
381 IF E<3000 AND M>700 THEN
 389 ELSE 382
382 IF E<3040 AND M>720 THEN
 389 ELSE 383
383 IF E<3060 AND M>740 THEN
 389 ELSE 384
384 IF E<3080 AND M>760 THEN
 389 ELSE 385
385 IF E<3120 AND M>780 THEN
 389 ELSE 386
386 IF E<3140 AND M>800 THEN
 389 ELSE 387
387 IF E<3160 AND M>820 THEN
 389 ELSE 398
388 STOP
389 CALL CLEAR :: CALL SCREE
N(10)
390 DISPLAY AT(3,1):"Sie H A
 B E N Anspruch auf  die Gew
aehrung von Wohngeld."
391 DISPLAY AT(10,1):"Fam.-E
inkommen: "|INT(E)|"DM"
392 DISPLAY AT(12,1):"anrech
enb. Miete: "|INT(M)|"DM"
393 FOR Q=1 TO 28 :: DISPLAY
 AT(6,Q):"*" :: CALL SOUND(7
5,Q*110,0):: NEXT Q
394 DISPLAY AT(20,1):"Falls
Sie eine neue Berech- nung w
uenschen, bitte eine  belieb
ige Taste druecken...."
395 CALL KEY(0,K,S):: IF S=0
 THEN 395
396 IF K THEN 65
397 END
398 CALL CLEAR :: CALL SCREE
N(10)
399 DISPLAY AT(3,1):"K E I N
  Wohngeldanspruch   -------
"
400 DISPLAY AT(10,1):"Fam.-E
```

LISTING

```
inkommen: "¦INT(E)¦"DM"
401 DISPLAY AT(12,1):"anrech
enb. Miete: "¦INT(M)¦"DM"
```

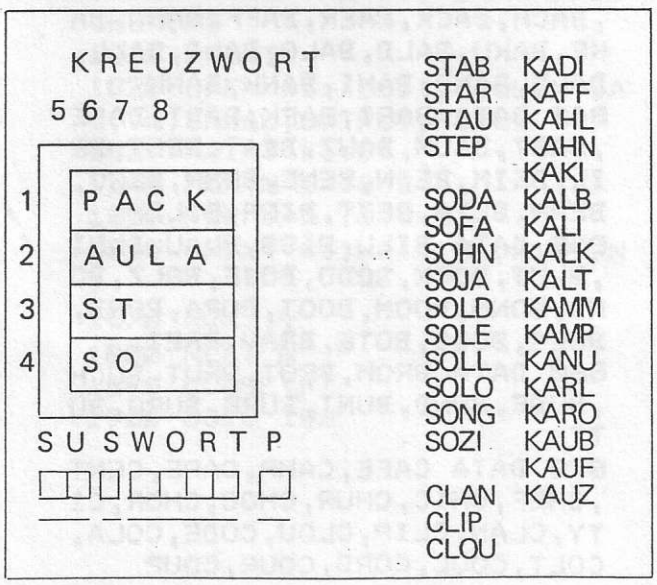
```
402 FOR Q=1 TO 28 :: DISPLAY
 AT(6,Q):"*" :: CALL SOUND(7
5,Q*110,0):: NEXT Q
403 GOTO 394
```

KREUZ-WORT-RÄTSEL

Ziel dieses Spiels ist es, quadratische Kreuzworträtsel zu erstellen. Dazu werden 8 Worte mit einer Länge von jeweils 4 Buchstaben in ein quadratisches Feld geschrieben. Jeder der diese Aufgabe zu lösen versucht, wird feststellen, daß dies nicht ganz einfach ist. Dabei sollte jeder auf seinen eigenen Wortschatz zurückgreifen. Da es oft genug vorkommt, daß einem selber kein passendes Wort einfällt, kann man den Computer befragen. Dieser verfügt über 1200 Worte, die absichtlich programmintern abgespeichert sind, damit ein umständliches Einlesen vom Cassettenrecorder entfällt.
Zum Suchen wird ein Suchausdruck von 2 Buchstaben eingegeben, woraufhin der Computer alle ihm bekannten Worte, welche die zwei Buchstaben an entsprechender Position enthalten, anzeigt.
Aus diesem Wortvorrat können Sie nun, durch geschickte Auswahl, Ihr Kreuzworträtsel vervollständigen, bis Sie ein komplettes Wort-Quadrat besitzen.

Bedienungsanleitung
Da im Programm auf jede Kommentierung aus Platzgründen verzichtet werden mußte, hier nun die Anleitung.

Bildschirm:



Beispiel:
Mit Auswahl der Worte STAR, SONG, KARG CLAN ist man am Ziel.

	WORT		PLATZ	
1) Eingabe Wort:	WERT (4B)	ENT	1 (1–8)	ENT

Wort wird in das Kreuzwortschema eingetragen.

	WORT		STELLE	
2) Eingabe Suchausdruck:	KA (2B)	ENT	1 (1–3)	ENT

Gesuchte Wörter werden auf dem Bildschirm angezeigt.

	WORT		PLATZ	
3) Zeile löschen:	L	ENT	3	ENT

Angegebene Zeile wird gelöscht.

	WORT		PLATZ	
4) Feld löschen:	L	ENT	A	ENT

Alle Wörter werden gelöscht.

Beispiele:

HEFE	OFEN	EDER	LESE
UFER	VERA	LIDO	IGEL
PELZ	ATOM	LEIS	REAL
EULE	LESE	ESTE	ALTE

Bedienungshinweise:
Die Taste Alpha-Lock muß eingerastet sein, da das Programm nur Großbuchstaben erkennt. Bei Eingabe eines Suchausdrucks wird für Stelle eine Ziffer zwischen 1 und 3 eingegeben. Dies entspricht den Positionen vorne, Mitte, hinten. Die Zeiten,die der Computer zum Suchen benötigt, sind, wenn man den Anfangsbuchstaben mit eingibt, recht kurz. Ohne Anfangsbuchstaben ist er gezwungen alle Worte abzusuchen, so daß es schon eine Weile dauert. Bedenkt man, daß,wenn man selber suchen wollte, man z.B. den gesamten Duden nach Worten durchforsten müßte, so nimmt man diese Wartezeit gerne in Kauf.
Die Zeiten betragen für Suchen mit Anfangsbuchstaben ca. 15 s und ohne ca. 2 min. Im Normalfall kommt man mit der schnellen Suchfunktion aus. Dennoch ist für spezielle Fälle die lange Suchfunktion von großem Vorteil.
Es sei noch darauf hingewiesen, daß das Programm die Speicherkapazität bis auf 185 Byte voll in Anspruch nimmt. Eine Verwendung dieses Restes, sei es für Kommentare oder Ergänzungen, ist nicht zu empfehlen, da dies die Suchgeschwindigkeit erheblich reduzieren würde.

```
10 ! ************************
11 ! *                      *
12 ! *     MAGISCHES        *
13 ! *  KREUZWORTRAETSEL    *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *   Harald Bertsch     *
17 ! *                      *
18 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    11670 Bytes       *
28 ! *                      *
29 ! ************************
30 !
180 CALL CLEAR
190 DIM F$(4,4)
200 ON WARNING NEXT
210 CALL MAGNIFY(2)
220 FOR I=3 TO 4
230 CALL COLOR(I,8,8)
240 CALL COLOR(I+6,8,8)
250 NEXT I
260 CALL CHAR(96,"1Ø1Ø1Ø1Ø1Ø
1Ø1Ø1ØØØØØØFFFFØØØØØØ1Ø1Ø1Ø
FFFF1Ø1Ø1Ø1Ø1ØFFFF")
270 CALL CHAR(1ØØ,"1Ø1Ø1Ø1F1
F1Ø1Ø1ØØØØØØFFFF1Ø1Ø1Ø1Ø1
ØFØFØ1Ø1Ø1ØØØØØØ1F1F1Ø1Ø1")
280 CALL CHAR(1Ø4,"ØØØØØØFØF
Ø1Ø1Ø1Ø1Ø1Ø1ØFØFØØØØØØØ1Ø1Ø1
Ø1F1FØØØØØØØØØØØØØØØØØØØØ"
)
290 CALL CHAR(128,"FFFFFFFFF
FFFFFFF3F3F3F3F3F3F3FFCFCF
CFCFCFCFCFCØØFF")
300 DISPLAY AT(1,6)SIZE(17):
"KREUZWORT"
310 FOR K=6 TO 18 :: CALL HC
HAR(K,9,1Ø7,7):: NEXT K
320 FOR K=6 TO 18 STEP 3 ::
CALL HCHAR(K,9,97,7):: NEXT
K
330 FOR K=8 TO 16 STEP 2 ::
CALL VCHAR(7,K,96,11):: NEXT
 K
340 CALL HCHAR(6,8,1Ø3)
350 CALL HCHAR(6,16,1Ø4)
360 CALL HCHAR(18,8,1Ø6)
370 CALL HCHAR(18,16,1Ø5)
380 FOR K=Ø TO 2
390 CALL HCHAR(6,1Ø+2*K,1Ø1)
400 CALL HCHAR(9,1Ø+2*K,98)
410 CALL HCHAR(12,1Ø+2*K,98)
420 CALL HCHAR(15,1Ø+2*K,98)
430 CALL HCHAR(18,1Ø+2*K,99)
440 NEXT K
450 FOR K=9 TO 15 STEP 3
460 CALL VCHAR(K,8,1ØØ)
470 CALL VCHAR(K,16,1Ø2)
480 NEXT K
```

LISTINGS

```
490 CALL HCHAR(5,7,128,11)
500 CALL HCHAR(19,7,128,11)
510 CALL VCHAR(5,7,129,15)
520 CALL VCHAR(5,17,13Ø,15)
530 FOR K=Ø TO 3
540 CALL HCHAR(3,K*2+9,53+K)
550 CALL VCHAR(7+K*3,5,49+K)
560 NEXT K
570 DISPLAY AT(21,5)SIZE(18)
:"SU S WORT P     "
580 CALL HCHAR(24,7,131,2)
590 CALL HCHAR(24,1Ø,131)
600 CALL HCHAR(24,12,131,4)
610 CALL HCHAR(24,17,131)
620 FOR I=3 TO 4
630 CALL COLOR(I,2,8)
640 CALL COLOR(I+6,2,16)
650 NEXT I
660 FOR Z=1 TO 4
670 FOR S=1 TO 4
680 F$(Z,S)=" "
690 NEXT S :: NEXT Z
700 CALL ANZ(NW$,PL,F$(,))
710 CALL EIN(NW$,PL,S$,L,PO)
720 IF L=Ø THEN 710
730 IF L=2 THEN 770
740 IF PL=Ø THEN 660
750 CALL ANZ(NW$,PL,F$(,))
760 IF L=4 OR L=1 THEN 710
770 CALL SUCH(S$,PO)
780 GOTO 710
790 END
800 DATA ABBA,ABEL,ABER,ACHT
,ACRE,ADAM,ADEN,ADEL,AEON,AE
RA,AFFE,AGIL,AHLE,AHNE,AIDA,
AIDS,AJAX,AKKU,AKNE,AKTE
810 DATA AKUT,ALEX,ALFA,ALGE
,ALMA,ALPE,ALSO,ALTE,AMEN,AM
ME,AMOK,AMOR,AMUR,ANAL,ANIS,
ANNA,ANNI,ANNO,ANTI,ANUS
820 DATA APER,APEX,AQUA,ARAL
,ARES,ARGE,ARME,ARNO,ASTI,AS
YL,ATEM,ATOM,AUCH,AULA,AURA,
AUTO,AVIS,AVUS,AXEL,AZID
830 DATA BAAL,BAAR,BAAS,BABY
,BACH,BACK,BAER,BAFF,BAHN,BA
KE,BAKU,BALD,BALG,BALI,BALL,
BALZ,BAND,BANI,BANK,BANN
840 DATA BARI,BARK,BART,BASE
,BAST,BAUM,BAUZ,BEAT,BEET,BE
IL,BEIM,BEIN,BENE,BENN,BERG,
BERN,BETA,BETT,BIER,BILD
850 DATA BILL,BISS,BLAU,BLEI
,BLUT,BOCK,BODO,BOJE,BOLZ,BO
ND,BONN,BOOM,BOOT,BORA,BORD,
BOES,BOSS,BOTE,BRAV,BREI
860 DATA BROM,BROT,BRUT,BUCH
,BUDE,BUND,BUNT,BURE,BURG,BU
TT
870 DATA CAFE,CAMP,CAPE,CENT
,CHEF,CHIC,CHUR,CHOC,CHOR,CI
TY,CLAN,CLIP,CLOU,CODE,COLA,
COLT,COOL,CORD,COUR,COUP
```

LISTINGS

```
880 DATA CREW,CRUX,CUBA,CURE
,CYAN
890 DATA DACH,DANK,DANN,DAME
,DARK,DARM,DASS,DATO,DAZU,DE
CK,DEIN,DEKA,DEPP,DERB,DEUT,
DEZI,DIAS,DICK,DIEB,DIES
900 DATA DILL,DING,DIRK,DITO
,DIVA,DOCH,DOCK,DOGE,DOLE,DO
LL,DOOF,DORA,DORF,DORN,DORT,
DOWN,DREH,DREI,DRUM,DUAL
910 DATA DUFT,DUMM,DUNE,DUNG
,DUST
920 DATA EARL,EBBE,EBEN,EBER
,EBRO,ECHO,ECHT,ECKE,EDAM,ED
DA,EDEL,EDEN,EDLE,EGAL,EGEL,
EGER,EGGE,EGON,EHER,EIBE
930 DATA EINE,EIRE,EISS,EKEL
,ELAN,ELBA,ELBE,ELKE,ELCH,EL
FE,ELFT,ELLE,ELLI,ELSE,EMIL,
EMIR,ENDE,ENTE,EPIK,ERBE
940 DATA ERGO,ERIS,ERLE,ERNA
,EROS,ESEL,ESPE,ESRA,ESSE,ES
SO,ESTE,ETAT,ETUI,ETWA,EUCH,
EUER,EULE,EURE,EXIL,EXOT
950 DATA FACE,FACH,FADE,FAHL
,FAIR,FAIT,FAKT,FALL,FALZ,FA
NG,FARM,FARN,FARO,FASE,FASS,
FAST,FATA,FAUL,FAUN,FAXE,FEG
E
960 DATA FEHL,FEIG,FEIN,FEIL
,FELD,FELL,FELS,FEME,FENN,FE
RN,FEST,FETE,FETT,FEWA,FIAT,
FIFA,FILM,FILZ,FINA,FINK,FIP
S,FIRN
970 DATA FLAK,FLIP,FLIT,FLOH
,FLOR,FLUG,FLUR,FLUT,FOEN,FO
ND,FORM,FORT,FOTO,FOUL,FRAU,
FREI,FROH,FRON,FUGE,FUND,FUN
K,FURZ
980 DATA GAEA,GABE,GABI,GAGE
,GALA,GANG,GANS,GANZ,GARE,GA
RN,GAST,GATE,GATT,GAUL,GAZE,
GEHE,GEHL,GEIL,GEIZ,GELB,GEL
D
990 DATA GELL,GEMA,GEMS,GENE
,GENF,GENT,GERD,GERN,GEST,GI
ER,GIFT,GIGA,GIPS,GIRL,GIRO,
GLAS,GOAL,GOBI,GODE,GOER,GOL
D,GOLF
1000 DATA GONG,GOSE,GOTE,GOT
T,GOYA,GRAB,GRAD,GRAF,GRAL,G
RAM,GRAN,GRAS,GRAU,GRAZ,GRIT
,GROB,GROG,GUPF,GURT,GURU,GU
SS,GUTE
1010 DATA HAAR,HABE,HACK,HAF
F,HAFT,HAHN,HAIN,HALB,HALL,H
ALO,HALS,HALT,HAND,HANF,HANG
,HANS,HARM,HARN,HART,HARZ,HA
SE
1020 DATA HASS,HAST,HATZ,HAU
E,HAUS,HAUT,HAXE,HECK,HEER,H
EFE,HEFT,HEGE,HEHL,HEIL,HEIM
,HEIN,HELD,HELL,HELM,HEMD
1030 DATA HERA,HERB,HERD,HER
O,HERR,HERZ,HEXA,HIEB,HIER,H
INZ,HIOB,HIRN,HIRT,HOCH,HODE
,HOFE,HOHE,HOHL,HOHN,HOLD
1040 DATA HOLM,HOLZ,HOMO,HOP
P,HORA,HORN,HUCH,HUFE,HUGO,H
UHN,HULD,HUND,HUNT,HUPE,HUPF
,HURE
1050 DATA IBIS,IDOL,IDUN,IGE
L,IGLU,ILEX,ILSE,IMAN,IMME,I
MUN,INCH,INGO,INKA,IOTA,IOWA
,IRAK,IRAN,IRIS,ISAR,ISER,IT
EM,IWAN
1060 DATA JACK,JADE,JAGD,JAH
N,JAHR,JANE,JAVA,JAZZ,JEAN,J
EEP,JEFF,JENS,JOCH,JOGA,JOGI
,JOHN,JUDE,JUDO,JULI,JUNG
1070 DATA JUNI,JUNO,JURA,JUR
Y,JUST,JUTE
1080 DATA KADI,KAFF,KAHL,KAH
N,KAKI,KALB,KALI,KALK,KALT,K
AMM,KAMP,KANU,KARL,KARO,KAUB
,KAUF,KAUZ,KECK,KEES,KEHL
1090 DATA KEIF,KEIL,KEIM,KEI
N,KEKS,KERL,KERN,KESS,KICK,K
IDS,KIEL,KIEN,KIES,KIEW,KILO
,KIND,KINN,KINO,KITT,KITZ
1100 DATA KIWI,KLAN,KLAR,KLE
E,KLUB,KLUG,KNIE,KOCH,KODE,K
OHL,KOKS,KOLA,KOMA,KORB,KORD
,KORK,KORN,KOST,KRAD,KRAM
1110 DATA KRAN,KRIM,KRUG,KUB
A,KUFE,KULE,KULI,KULT,KURS,K
URT,KURZ,KUSS
1120 DATA LACK,LADY,LAGE,LAH
N,LAHR,LAIB,LAIE,LAKE,LAMA,L
AMM,LAND,LANG,LAOS,LAST,LATZ
,LAUB,LAUF,LAUS,LAUT,LAVA
1130 DATA LEER,LECK,LEIB,LEI
D,LEIM,LENA,LENZ,LESE,LIAS,L
IDO,LIEB,LIED,LIES,LIFT,LIGA
,LILA,LIMA,LINK,LINZ,LIRA
1140 DATA LIRE,LIST,LOCH,LOG
E,LOHN,LOLA,LORD,LORE,LOSE,L
OVE,LUFT,LULU,LUMP,LUNA,LUPE
,LUST,LUTZ,LYRA
1150 DATA MAAR,MAAS,MAAT,MAE
H,MADE,MAGD,MAHL,MAHR,MAID,M
AIN,MAIS,MALZ,MAMA,MANN,MARK
,MARS,MARX,MARY,MASS,MAST
1160 DATA MATT,MAUL,MAUS,MAU
T,MAYA,MEER,MEGA,MEHL,MEHR,M
EIN,METT,MIAU,MICH,MIEF,MIES
,MILD,MILZ,MIME,MINE,MISS
1170 DATA MODE,MOFA,MOHN,MOH
R,MOLE,MOLL,MOND,MONO,MOOR,M
OOS,MOPS,MORD,MOST,MUEH,MUFF
,MULL,MUMM,MUND,MUSE,MUSS
1180 DATA NABE,NACH,NAHE,NAI
V,NAME,NANO,NANU,NAPF,NARR,N
ASE,NASS,NATO,NAZI,NEID,NEIN
,NERV,NERZ,NEST,NETT,NETZ
1190 DATA NEUE,NEUN,NIET,NIN
```

»

```
A,NOAH,NOCH,NONE,NORD,NORM,N
OTE,NOVA,NULL,NUSS,NUTZ
1200 DATA OBEN,OBER,OBIG,OBO
E,OBST,OCHS,ODAL,ODEM,ODER,O
ERE,OESE,OHIO,OHNE,OKAY,OKER
,OLAF,OLGA,OLIM,OLIV,OMAR
1210 DATA OMEN,ONAN,ONYX,OPA
K,OPAL,OPEL,OPER,OPUS,ORAL,O
RFE,OSLO,OVAL,OXER,OXYD,OXID
,OZON
1220 DATA PAAR,PACK,PAFF,PAG
E,PAIR,PAKT,PAPA,PARA,PARD,P
ARK,PART,PASS,PATE,PAUL,PECH
,PEIN,PELZ,PEPI,PERI,PERL
1230 DATA PERM,PERU,PESO,PES
T,PFAD,PFAU,PHON,PIEP,PIKE,P
INK,PIPE,PISS,PLAN,PNEU,POCH
,POEM,POET,POLE,POLO,POMP
1240 DATA POND,PONY,POOL,POP
E,POPO,PORE,PORT,POSE,POST,P
OTT,PRAG,PUCK,PUFF,PULK,PULT
,PUMA,PUMP,PUTZ
1250 DATA QUAI,QUAK,QUAL,QUA
S,QUJE,QUER,QUIT,QUIZ
1260 DATA RAAB,RADE,RADI,RAF
T,RAGE,RAHE,RAHM,RAIN,RALF,R
AND,RANG,RANK,RAPS,RATE,RATZ
,RAUB,RAUH,RAUM,RAUS,REAL,RE
BE,RECK
1270 DATA REDE,REGE,REIF,REI
M,REIN,REIS,REIZ,RENE,REUE,R
IAL,RIAS,RIED,RIFF,RIGA,RIGI
,RIND,RING,RISS,RITA,RITE
1280 DATA RITZ,ROBE,ROCK,ROH
R,ROOF,ROSL,ROSS,ROST,ROTZ,R
OUT,RUCH,RUCK,RUDI,RUFE,RUHM
,RUHR,RUIN,RUND,RUNE,RUSS,RU
TE,RUTH
1290 DATA SAAL,SAAR,SAAT,SAB
A,SACK,SAFE,SAFT,SAGA,SALZ,S
AME,SAMT,SAND,SANG,SARA,SASS
,SATT,SATZ,SAUF,SAUM,SCHI
1300 DATA SEIL,SEIN,SEIT,SEK
T,SEMI,SENF,SENN,SHAG,SHOW,S
IAM,SICH,SIEB,SIEG,SIEH,SIEL
,SIKH,SILK,SILO,SIMS,SING
1310 DATA SINN,SIRE,SITZ,SKA
T,SLIP,SLUM,SMOG,SNOB,SODA,S
OFA,SOHN,SOJA,SOLD,SOLE,SOLL
,SOLO,SONG,SOZI,SPAN,SPAR
1320 DATA SPAT,SPOR,SPUK,SPU
R,STAB,STAR,STAU,STEP,STIL,S
TOP,STUR,SUED,SUEZ,SUFF,SUND
,SYLT
1330 DATA TABU,TAKT,TALG,TAN
D,TANG,TANK,TANZ,TARA,TAUB,T
AXE,TAXI,TEAM,TEEN,TEER,TEIG
,TEIL,TEIN,TELE,TERM,TERZ
1340 DATA TEST,TETE,TEXT,THE
A,THEO,TICK,TIEF,TIER,TILL,T
ODE,TOGO,TOLL,TONI,TOPF,TOPP
,TORF,TORR,TOTE,TOTO,TOUR
1350 DATA TRAB,TRAM,TRAN,TRI
P,TRUG,TUBA,TUBE,TUCH,TUFF,T
UPF,TUER,TURN,TWEN,TYPE
1360 DATA UFER,UKAS,ULAN,ULL
A,ULME,UNKE,UNZE,URAL,URAN,U
RAT,URBE,URFT,URIG,URIN,URNE
,UTAH
1370 DATA VALE,VAMP,VASE,VAT
I,VELO,VENE,VENN,VERA,VERB,V
ERS,VETO,VIEL,VIER,VISA,VIZE
,VOGT,VOLK,VOLL,VOLT,VOTA
1380 DATA WAAL,WABE,WACH,WAD
E,WADI,WAHL,WAHN,WAHR,WALD,W
ALL,WAMS,WAND,WARE,WARM,WART
,WASA,WATT,WECK,WEGA,WEHE
1390 DATA WEHR,WEIB,WEIL,WEI
N,WEIT,WELK,WELS,WELT,WENN,W
ERK,WERT,WEST,WIEN,WILD,WIND
,WINK,WIRR,WIRT,WITZ
1400 DATA WOHL,WOLF,WORT,WUN
D,WUNE,WURF,WURM,WURZ,WUST
1410 DATA XERO,XMAL,XRAY,YBB
S,YSOP,YARD,YAWL,YEAR,YETI,Y
UAN
1420 DATA ZACK,ZAEH,ZAHL,ZAH
M,ZAHN,ZANK,ZAPF,ZART,ZAUM,Z
AUN,ZECK,ZEHN,ZEIT,ZELT,ZERO
,ZETA,ZEUG,ZEUS,ZIEL,ZIER
1430 DATA ZIMT,ZINK,ZINN,ZIN
S,ZION,ZIPF,ZOLA,ZOLL,ZOPF,Z
ORN,ZOTE,ZUCK,ZULU,ZWAR,ZWEI
,ZYAN
1440 SUB ANZ(NW$,PL,F$(,))
1450 IF NW$="L" THEN NW$="
  "
1460 IF PL>0 AND PL<5 THEN 1
480
1470 IF PL>4 AND PL<9 THEN 1
490 ELSE 1500
1480 Z=PL :: FOR S=1 TO 4 ::
 F$(Z,S)=SEG$(NW$,S,1):: NEX
T S :: GOTO 1500
1490 S=PL-4 :: FOR Z=1 TO 4
:: F$(Z,S)=SEG$(NW$,Z,1):: N
EXT Z
1500 FOR Z=0 TO 3 :: FOR S=1
 TO 4
1510 CALL SPRITE(#S+Z*4,ASC(
F$(Z+1,S)),2,48+24*Z,45+16*S
)
1520 NEXT S :: NEXT Z
1530 SUBEND
1540 SUB EIN(NW$,PL,S$,L,PO)
1550 ACCEPT AT(23,10)SIZE(4)
VALIDATE(UALPHA,"L"):NW$
1560 L=LEN(NW$)
1570 IF NW$="L" THEN 1610
1580 IF L=0 THEN 1700
1590 IF L=2 THEN 1650
1600 IF L<>4 THEN 1550
1610 ACCEPT AT(23,15)SIZE(1)
VALIDATE(DIGIT,"A"):PL$
1620 IF PL$="A" THEN PL=0 ::
 GOTO 1700
```

```
1630 PL=VAL(PL$)
1640 IF PL=0 OR PL=9 THEN 16
10 ELSE 1700
1650 S$=NW$
1660 DISPLAY AT(23,5)SIZE(3)
:S$
1670 DISPLAY AT(23,10)SIZE(3
):"  "
1680 ACCEPT AT(23,8)SIZE(1)V
ALIDATE(DIGIT):PO
1690 IF PO=0 OR PO>3 THEN 16
80
1700 SUBEND
1710 SUB SUCH(S$,PO)
1720 DIM W$(130)
1730 IF PO>1 THEN 1770
1740 Y=ASC(S$)-64
1750 IF Y>20 THEN 1870
1760 ON Y GOTO 1790,1790,180
0,1800,1800,1810,1810,1820,1
820,1820,1820,1830,1830,1840,1840
,1840,1850,1850,1850,1860,18
60
1770 FOR K=1 TO 9
1780 ON K GOTO 1790,1800,181
0,1820,1830,1840,1850,1860,1
870
1790 RESTORE 800 :: GOTO 188
0
1800 RESTORE 870 :: GOTO 188
0
1810 RESTORE 950 :: GOTO 188
0
1820 RESTORE 1010 :: GOTO 18
80
1830 RESTORE 1080 :: GOTO 18
80
1840 RESTORE 1150 :: GOTO 18
80
1850 RESTORE 1220 :: GOTO 18
80
1860 RESTORE 1290 :: GOTO 18
80
1870 RESTORE 1360 :: GOTO 18
80
1880 FOR I=1 TO 130 :: READ
W$(I)
1890 IF S$=SEG$(W$(I),PO,2)T
HEN J1=J1+1 :: IF J1<25 THEN
 DISPLAY AT(J1,20)SIZE(5):W$
(I)ELSE DISPLAY AT(J1-24,25)
SIZE(5):W$(I)
1900 IF J1=48 THEN J1=0
1910 NEXT I
1920 IF PO=1 THEN 1940
1930 NEXT K
1940 J1=J1+1
1950 IF J1<25 THEN CALL HCHA
R(J1,21,31,5)ELSE CALL HCHAR
(J1-24,26,31,5)
1960 SUBEND
```

COMPUTER-TITEL AUS DEM VERLAG

LISTINGS

```
10 REM*********************
11 REM*                    *
12 REM* FUENFZEHNERSPIEL  *
13 REM*  Copyright  by     *
14 REM* Burkhardt Knedel  *
15 REM*                    *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole  *
19 REM*  Joystick (1)      *
20 REM*                    *
23 REM* Speicherbelegung  *
24 REM*    2778 Bytes      *
25 REM*********************
26 REM
100 CALL CLEAR
110 CALL SCREEN(12)
120 CALL COLOR(13,5,5)
130 RANDOMIZE
140 CALL CHAR(128,"FFFFFFFFF
FFFFFFF")
150 V$="VERSUCH:"
160 S$="FUENFZEHNERSPIEL"
170 PRINT S$: : : : :"SIE MU
ESSEN 15 ZAHLEN ORDNEN": : :
 : :
180 PRINT "BEWEGEN SIE DAS L
EERFELD": :"ENTSPRECHEND DER
 ZAHLEN MIT": :"DER FERNBEDI
ENUNG!": : : : :
190 PRINT "DRUECKEN SIE DEN
ROTEN KNOPF"
200 CALL KEY(1,X,S)
210 IF X<>18 THEN 200
220 CALL VCHAR(1,1,32,768)
230 FOR I=1 TO LEN(S$)
240 CALL SOUND(-30,110,1)
250 CALL HCHAR(1,8+I,ASC(SEG
$(S$,I,1)))
260 NEXT I
270 FOR I=3 TO 19 STEP 4
280 CALL SOUND(-30,-6,1)
290 CALL HCHAR(I,9,128,17)
300 NEXT I
310 FOR I=9 TO 25 STEP 4
320 CALL SOUND(-30,-6,1)
330 CALL VCHAR(3,I,128,17)
340 NEXT I
350 DIM R(19,25)
360 FOR Z=5 TO 17 STEP 4
370 FOR S=11 TO 23 STEP 4
380 CALL SOUND(-30,-3,1)
390 A=A+1
400 IF A<16 THEN 420
410 A=0
420 R(Z,S)=A
430 A$=STR$(A)
440 FOR J=1 TO LEN(A$)
450 CALL HCHAR(Z,S-1+J,ASC(S
EG$(A$,J,1)))
460 NEXT J
470 NEXT S
480 NEXT Z
490 O$="SO SOLL ES AUSSEHEN!"
500 FOR I=1 TO LEN(O$)
510 CALL HCHAR(24,7+I,ASC(SE
G$(O$,I,1)))
520 NEXT I
530 FOR I=1 TO 1500
540 NEXT I
550 CALL HCHAR(24,1,32,32)
560 FOR J=1 TO 20
570 Z=INT((RND*4)+1)*4+1
580 S=INT((RND*4)+1)*4+7
590 T=R(Z,S)
600 X=INT((RND*4)+1)*4+1
610 Y=INT((RND*4)+1)*4+7
620 R(Z,S)=R(X,Y)
630 R(X,Y)=T
640 CALL HCHAR(Z,S,32,2)
650 CALL SOUND(-30,777,1)
660 A$=STR$(R(Z,S))
670 FOR I=1 TO LEN(A$)
680 CALL HCHAR(Z,S-1+I,ASC(S
EG$(A$,I,1)))
690 NEXT I
700 CALL HCHAR(X,Y,32,2)
710 CALL SOUND(-30,777,1)
720 A$=STR$(R(X,Y))
730 FOR I=1 TO LEN(A$)
740 CALL HCHAR(X,Y-1+I,ASC(S
EG$(A$,I,1)))
750 NEXT I
760 NEXT J
770 FOR Z=5 TO 17 STEP 4
780 FOR S=11 TO 23 STEP 4
790 IF R(Z,S)=0 THEN 820
800 NEXT S
810 NEXT Z
820 CALL HCHAR(Z,S,32,2)
830 ZZ=Z
840 SS=S
850 FOR I=1 TO LEN(V$)
860 CALL SOUND(-50,440,1,660
,1)
870 CALL HCHAR(22,10+I,ASC(S
EG$(V$,I,1)))
880 NEXT I
890 CALL JOYST(1,XX,YY)
900 IF (XX<>0)+(YY<>0)=-2 TH
EN 890
910 IF (XX=0)+(YY=0)=-2 THEN
 890
920 CALL SOUND(300,110,11,44
0,11,880,11)
930 P=P+1
940 IF XX=4 THEN 1050
950 IF XX=-4 THEN 1090
960 IF YY=-4 THEN 1010
970 Z=Z+4
980 IF Z<=17 THEN 1130
990 Z=Z-4
1000 GOTO 890
1010 Z=Z-4
1020 IF Z>=5 THEN 1130
1030 Z=Z+4
1040 GOTO 890
```

```
1050 S=S-4
1060 IF S>=11 THEN 1130
1070 S=S+4
1080 GOTO 890
1090 S=S+4
1100 IF S<=23 THEN 1130
1110 S=S-4
1120 GOTO 890
1130 P$=STR$(P)
1140 FOR I=1 TO LEN(P$)
1150 CALL HCHAR(22,18+I,ASC(
SEG$(P$,I,1)))
1160 NEXT I
1170 CALL HCHAR(Z,S,32,2)
1180 A$=STR$(R(Z,S))
1190 FOR I=1 TO LEN(A$)
1200 CALL HCHAR(ZZ,SS-1+I,AS
C(SEG$(A$,I,1)))
1210 NEXT I
1220 T=R(Z,S)
1230 R(Z,S)=R(ZZ,SS)
1240 R(ZZ,SS)=T
1250 FOR U=5 TO 17 STEP 4
1260 FOR V=11 TO 23 STEP 4
1270 W=W+1
1280 IF W<=15 THEN 1300
1290 W=0
1300 IF R(U,V)<>W THEN 1340
1310 NEXT V
1320 NEXT U
1330 GOTO 1380
1340 ZZ=Z
1350 SS=S
1360 W=0
1370 GOTO 890
1380 CALL CLEAR
1390 PRINT "HERZLICHEN GLUEC
KWUNSCH!": : : :"SIE HABEN E
S MIT";P;: : : :"VERSUCHEN G
ESCHAFFT!": : : : : :
1400 PRINT "AUF WIEDERSEHEN.
": : :
```

FÜNFZEHNERSPIEL

Wer kennt sie nicht, die kleinen Geduldsspiele für längere Pausen oder Wartezeiten.
Ein kleines Viereck mit Zahlen oder Bildern, die etwas durcheinandergeraten sind.
Genauso durcheinandergeraten sind dem Autor dieses Programmes seine 15 Zahlen auf dem Bildschirm. Mit Hilfe des Joysticks und eines Leerfeldes sollten Sie die fortlaufende Numerierung von oben links nach unten rechts eigentlich schnell wieder hinbekommen.
Obwohl das Programm auch in TI-Basic läuft, reagieren die einzelnen Zahlenfelder sofort auf jede Joystickbewegung. Lange Wartezeiten, bis der Computer die neue Stellung berechnet hat, fallen damit flach.
Sie werden hundertprozentig länger zum Überlegen benötigen, als der TI zum Darstellen der neuen Positionen.
Unter dem Quadrat wird gleichzeitig die Anzahl der Versuche angezeigt.
Einem Wettbewerb unter Freunden steht demnach nichts mehr im Wege.

COMPUTER-TITEL
AUS DEM VERLAG

FLASHDANCE

Bei Flashdance handelt es sich um ein Gedächtnisspiel, welches an den Spieler eine ganz besondere Herausforderung stellt. Das Spiel zeichnet sich vor allem durch seine hervorragende Grafik aus. Dabei werden 'Multicolorsprites' und andere Tricks benutzt.
Du befindest Dich in einer Großstadt. Breakdance ist wieder in. Da kommt auch schon ein Herausforderer und will sich mit Dir messen. Du bist der Champ! Zeig's ihm.
Zunächst sieht man ein kleines Demo. Das Spiel wird über den Joystick gesteuert. Bei Knopfdruck startet dann das eigentliche Spiel. Der Herausforderer (links im Bild) macht Dir eine Bewegung vor, diese sollst Du nachmachen. Aber laß Dir nicht zu lange Zeit, sonst wirst Du ausgezählt. Machst Du die Bewegung richtig nach, geht es weiter. Jetzt kommt eine neue Bewegung hinzu und wieder bist Du an der Reihe. Maximal werden 50 Bewegungen vorgemacht, dann ist das Spiel zu Ende.
Bei Überschreiten des Zeitlimits oder einer falschen Bewegung ist ebenfalls Schluß.
Der Highscore wird unter dem linken Männchen angezeigt.
Die Punkte errechnen sich aus verwendeter Zeit und Anzahl der Bewegungen.

Steuerung:

Knopf:	Sprung nach oben
oben:	Arme hoch
unten:	Rolle
links:	Schrittstellung
rechts:	Hut ab

CYCLONS

Bei dem Spiel "Cyclons" handelt es sich um ein Action-Spiel für zwei Personen. Benötigt werden das Extended-Basic-Modul und zwei Joysticks.
Nach dem Verschwinden des Titelbildes kann man sich für einen von 4 Schwierigkeitsgraden entscheiden. Daraufhin wird um die Eingabe der gewünschten Anzahl von Raumschiffen gebeten (Eingabe mit ENTER abschließen).
Nun beginnt das Spiel. Es erscheinen vor einem Sternenhintergrund zwei Raumschiffe, von denen das linke dem Spieler mit Joystick 1 und das rechte dem Spieler mit Joystick 2 gehört. Ziel des Spieles ist es, durch geschickte Lenkmanöver das eigene Raumschiff zu bewahren und das andere Raumschiff durch einen geschickten Schuß zu zerstören.
Geschossen werden kann durch Halten des Joysticks in die gewünschte Richtung und gleichzeitiges Drükken des Feuerknopfes. Hat ein Raumschiff einen Schuß abgegeben, wird es transparent und kann weder getroffen noch gerammt werden, solange der Schuß sichtbar ist.
Interessant und auch erst amüsant wird das Spiel durch die seltsame Steuerung und die Möglichkeit, aus dem Bildschirm an einer Seite herauszufliegen, um sich hinter dem Bildrand zu verstecken oder an der gegenüberliegenden Seite wieder hervorzukommen.
Alles in allem wünsche ich viel Spaß bei diesem Spiel.

Matthias Renken

```
10 ! ************************
11 ! *                      *
12 ! *     FLASHDANCE       *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *     Dirk Biehl       *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *    Joystick (1)      *
23 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    12014 Bytes       *
28 ! *                      *
29 ! ************************
30 !
100 ! (c) 1983 by G*B*S
110 CALL SCREEN(1):: CALL ST
ADT :: CALL SCREEN(8):: CALL
 MAGNIFY(4):: DIM X(50),Y(50
):: CALL GRUND
120 CALL MANN1 :: CALL DEMO
:: FOR I=1 TO 5 :: CALL FARB
130 CALL SPRITE(#1,128,14,11
5,50,#3,132,5,147,50,#2,136,
12,163,50,#4,140,2,83,50)::
FOR Q=1 TO 100 :: NEXT Q
140 ON I GOSUB 450,470,490,5
10,560
150 NEXT I
160 CALL SPRITE(#1,128,14,11
5,50,#3,132,5,147,50,#2,136,
12,163,50,#4,140,2,83,50)::
FOR Q=1 TO 100 :: NEXT Q
170 MERK=1 :: ZAHL=10 :: CAL
L CLEAN :: CALL DELSPRITE(AL
L):: RANDOMIZE :: CALL HCHAR
(19,15,32,3)
180 FOR I=1 TO 50 :: X(I)=IN
T(RND*5)+1 :: NEXT I :: CALL
 PUNKTE(0,0)
190 CALL SPRITE(#1,128,14,11
5,50,#3,132,5,147,50,#2,136,
12,163,50,#4,140,2,83,50)
200 CALL SPRITE(#11,128,9,11
5,150,#13,132,13,147,150,#12
,136,10,163,150,#14,140,5,83
,150)
210 CALL KEY(1,K,S):: IF S=0
 THEN 210
220 CALL HCHAR(24,19,32,6)
230 CALL DELSPRITE(#11,#12,#
13,#14):: CALL MANN1
240 DISPLAY AT(16,9)SIZE(2):
"XY" :: FOR I=1 TO 300 :: NE
XT I :: DISPLAY AT(16,9)SIZE
(2):"  "
250 FOR GBS=1 TO MERK :: CAL
L FARB
260 ON X(GBS)GOSUB 450,470,4
90,510,560
270 CALL SPRITE(#1,128,14,11
5,50,#3,132,5,147,50,#2,136,
12,163,50,#4,140,2,83,50)::
FOR Q=1 TO 25 :: NEXT Q
280 NEXT GBS :: CALL CLEAN :
: CALL DELSPRITE(#1,#2,#3,#4
):: CALL MANN2
290 CALL SPRITE(#1,128,14,11
5,50,#3,132,5,147,50,#2,136,
12,163,50,#4,140,2,83,50)::
FOR Q=1 TO 25 :: NEXT Q
300 DISPLAY AT(16,16):"ZI" :
: FOR I=1 TO 300 :: NEXT I :
: DISPLAY AT(16,16):"  "
310 FOR DB=1 TO MERK :: CALL
 FARB :: FOR Q=1 TO 25 :: NE
XT Q
320 FOR Q=1 TO 10 :: CALL KE
Y(1,K,S):: CALL JOYST(1,W,S)
:: IF ABS(W)=4 OR ABS(S)=4 O
R K=18 THEN 350 ELSE 330
330 NEXT Q
340 CALL ZAEHL(ZAHL):: IF ZA
HL<1 THEN 790 ELSE GOTO 320
350 CALL SOUND(50,110,0):: I
F K=18 THEN Y(DB)=4
360 IF W=-4 THEN Y(DB)=3
370 IF W=4 THEN Y(DB)=2
380 IF S=-4 THEN Y(DB)=5
390 IF S=4 THEN Y(DB)=1
400 ON Y(DB)GOSUB 620,640,66
0,680,730
410 CALL SPRITE(#11,128,9,11
5,150,#13,132,13,147,150,#12
,136,10,163,150,#14,140,5,83
,150):: IF X(DB)<>Y(DB)THEN
800
420 NEXT DB
430 CALL HCHAR(24,29,32):: C
ALL GOOD :: CALL PUNKTE(MERK
,ZAHL):: MERK=MERK+1 :: ZAHL
=10 :: IF MERK=51 THEN 810
440 CALL CLEAN :: CALL SPRIT
E(#1,128,14,115,50,#3,132,5,
147,50,#2,136,12,163,50,#4,1
40,2,83,50):: GOTO 230
450 CALL ARME
460 CALL PATTERN(#1,124,#3,1
20):: CALL SPRITE(#4,116,2,1
09,50):: FOR Q=1 TO 100 :: N
EXT Q :: RETURN
470 CALL HUT
480 CALL PATTERN(#1,124,#3,1
16):: CALL SPRITE(#4,120,2,1
13,82):: FOR Q=1 TO 100 :: N
EXT Q :: RETURN
490 CALL HAMPEL
500 CALL SPRITE(#4,120,2,109
,50):: CALL PATTERN(#1,124,#
3,116,#2,112):: FOR Q=1 TO 1
00 :: NEXT Q :: RETURN
510 CALL SPRUNG
520 CALL SPRITE(#1,128,14,95
,50,#3,132,5,127,50,#2,136,1
2,143,50,#4,140,2,63,50):: F
```

```
OR Q=1 TO 25 :: NEXT Q
530 CALL PATTERN(#1,124,#3,1
20):: CALL SPRITE(#5,116,5,1
11,18,#6,112,5,111,82,#2,108
,12,127,50):: FOR Q=1 TO 40
:: NEXT Q
540 CALL DELSPRITE(#5,#6)
550 CALL SPRITE(#1,128,14,95
,50,#3,132,5,127,50,#2,136,1
2,143,50,#4,140,2,63,50):: F
OR Q=1 TO 25 :: NEXT Q :: RE
TURN
560 CALL ROLLE
570 CALL DELSPRITE(#1,#2,#3,
#4)
580 CALL SPRITE(#1,124,14,14
7,50,#2,120,5,163,50,#4,140,
2,115,50):: FOR Q=1 TO 30 ::
 NEXT Q
590 CALL SPRITE(#1,116,14,14
7,50,#4,112,2,147,50,#2,108,
12,147,50):: FOR Q=1 TO 30 :
: NEXT Q
600 CALL PATTERN(#1,104,#4,1
00,#2,96):: FOR Q=1 TO 30 ::
 NEXT Q
610 CALL SPRITE(#1,124,14,14
7,50,#2,120,5,163,50,#4,140,
2,115,50):: FOR Q=1 TO 30 ::
 NEXT Q :: RETURN
620 CALL ARME
630 CALL PATTERN(#11,124,#13
,120):: CALL SPRITE(#14,116,
5,109,150):: FOR Q=1 TO 100
:: NEXT Q :: RETURN
640 CALL HUT
650 CALL PATTERN(#11,124,#13
,116):: CALL SPRITE(#14,120,
5,113,182):: FOR Q=1 TO 100
:: NEXT Q :: RETURN
660 CALL HAMPEL
670 CALL SPRITE(#14,120,5,10
9,150):: CALL PATTERN(#11,12
4,#13,116,#12,112):: FOR Q=1
 TO 100 :: NEXT Q :: RETURN
680 CALL SPRUNG
690 CALL SPRITE(#11,128,9,95
,150,#13,132,13,127,150,#12,
136,10,143,150,#14,140,5,63,
150):: FOR Q=1 TO 25 :: NEXT
 Q
700 CALL PATTERN(#11,124,#13
,120):: CALL SPRITE(#15,116,
13,111,118,#16,112,13,111,18
2,#12,108,10,127,150):: FOR
Q=1 TO 40 :: NEXT Q
710 CALL DELSPRITE(#15,#16)
720 CALL SPRITE(#11,128,9,95
,150,#13,132,13,127,150,#12,
136,10,143,150,#14,140,5,63,
150):: FOR Q=1 TO 25 :: NEXT
 Q :: RETURN
730 CALL ROLLE
```

LISTINGS

```
740 CALL DELSPRITE(#11,#12,#
13,#14)
750 CALL SPRITE(#11,124,9,14
7,150,#12,120,13,163,150,#14
,140,5,115,150):: FOR Q=1 TO
 30 :: NEXT Q
760 CALL SPRITE(#11,116,9,14
7,150,#14,112,5,147,150,#12,
108,10,147,150):: FOR Q=1 TO
 30 :: NEXT Q
770 CALL PATTERN(#11,104,#14
,100,#12,96):: FOR Q=1 TO 30
 :: NEXT Q
780 CALL SPRITE(#11,124,9,14
7,150,#12,120,13,163,150,#14
,140,5,115,150):: FOR Q=1 TO
 30 :: NEXT Q :: RETURN
790 CALL TIME :: GOTO 170
800 CALL HCHAR(24,29,32):: C
ALL WRONG :: GOTO 170
810 CALL CLEAN :: CALL DELSP
RITE(ALL):: CALL GOOD :: GOT
O 120
820 DATA EE00EEAAEE00EEAA,00
0000F2020202E2A,1F0A0AFF0000E
EAA,000000FF0000EEAA,2E202E2
A2E202E2A,2E202E2A2E20AEAA
830 DATA AEA0AEAAAEA0AEAA,AE
A0AEAAAEA0A0BF,EE001B2424242
4FF,EE000E8A8E8080FF,000000B
FA0A0AEAA,0000000808080808080
840 DATA 8080808080808080,80
8080BFA0A0AEAA,EE00EEAAEE000
0FF
850 DATA 7E7E666666667E7E,18
1818181818181818,7E7E061E78607
E7E,7E7E061E1E067E7E,6666667
E7E060606,7E7E60781E067E7E
860 DATA 7E7E607E7E667E7E,7E
7E0606060606,7E7E667E7E667
E7E,7E7E667E7E067E7E
870 DATA 003F2020202027200,00
FF0000789710100,00FF000040801
828,00FF0000000000000,00FF000
0000060A0,00FC040404040404
880 DATA 2020202020222221,10
F22C2023354880,50B0C38D12B34
C00,0102122B2A4E9200,2040888
7805020000,0404040484040404
890 DATA 2020202020202020,00
0000000000101000,0400853B52979
9C6,0000008A35498A11,0000112
A43426598,0404848424448404
900 DATA 2020F000000FF00,00
00FF182442FF00,0000FF000000F
F00,0404FC000000FF00
910 DATA 003C243C00003C24,3C
00003C243C00FF,0018180000181
800,0018242424424FF,FF81819
9998181FF,2E202EAAAEA0AEAA,E
E00EEAAAEA0A0FF,2E202E2A2E20
203F
920 DATA 008E8A8A8AEE0000,00
```

```
E9AAACAAE90000,0087828282820
000,0075557262520000
930 DATA 32,32,32,32,34,36,4
4,32,32,32,32,32,32,32,32,32
,32,32,32,32,32,32,32,32,32,
32,32,32
940 DATA 32,32,32,32,37,33,4
5,32,32,32,32,32,32,32,32,32
,32,32,34,35,36,43,35,36,44,
32,32,32
950 DATA 32,32,32,32,37,33,4
5,32,32,84,32,32,32,32,32,32
,32,32,37,33,33,39,33,33,45,
32,32,32
960 DATA 32,32,32,32,37,33,4
5,32,84,82,84,32,58,59,60,61
,62,63,37,33,33,39,33,33,45,
32,32,32
970 DATA 32,32,32,32,37,33,4
5,84,82,82,82,84,64,65,66,67
,68,69,37,33,33,39,33,33,46,
35,35,36
980 DATA 34,35,36,35,85,33,4
5,84,80,80,80,84,70,71,72,73
,74,75,37,33,33,39,33,33,39,
33,33,33
990 DATA 37,33,33,33,39,33,4
5,84,81,81,81,84,76,77,78,78
,77,79,38,33,33,39,33,33,39,
33,33,33
1000 DATA 37,33,33,33,39,33,
45,84,80,80,80,84,80,80,82,8
0,80,80,39,33,33,39,33,33,39
,33,33,33
1010 DATA 87,86,86,86,40,47,
84,84,81,81,81,84,81,81,83,8
1,81,81,40,41,42,40,41,42,40
,47,41,42
1020 SUB STADT
1030 CALL CLEAR :: RESTORE 8
20 :: FOR I=33 TO 91 :: READ
 A$ :: CALL CHAR(I,A$):: NEX
T I
1040 FOR I=2 TO 10 :: FOR J=
2 TO 29 :: READ B :: CALL HC
HAR(I,J,B):: NEXT J :: NEXT
I
1050 CALL VCHAR(6,30,44):: C
ALL VCHAR(7,30,45,4):: SUBEN
D
1060 SUB GRUND
1070 CALL CHAR(128,"030507ØD
Ø603Ø1Ø11F3F7C6764676467CØAØ
EØB060CØ8Ø8ØF8FC3EE626E626E6
")
1080 CALL CHAR(132,"76776707
Ø7Ø6Ø6Ø6Ø6ØCØCØCØC1C3CØØ6EEE
E6E0E0606060603030303038 3C00
")
1090 CALL CHAR(136,"ØCØCØCØC
ØØØØØØØØØØØØØØØØØØØØØØØØ3Ø3Ø
3Ø3ØØØØØØØØØØØØØØØØØØØØØØØØ
")
```

LISTINGS

```
1100 CALL CHAR(140,"ØØØØØØØØ
ØØØØØØØØØØØØØØØØØØØ3Ø3ØFØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØCØCØFØ
"):: SUBEND
1110 SUB ARME
1120 CALL CHAR(124,"Ø305076E
656361713F3F1CØ704Ø704Ø7CØAØ
EØ76A6C6868EFCFC38EØ2ØEØ2ØEØ
")
1130 CALL CHAR(120,"Ø6Ø7Ø7Ø7
Ø7Ø6Ø6Ø6Ø6ØCØCØCØC1C3CØØ6ØEØ
EØEØE0606060603030303038 3C00
")
1140 CALL CHAR(116,"Ø303ØF4Ø
6Ø6ØØØØØØØØØØØØØØØØØØØØØCØCØ
FØØ2Ø6Ø6ØØØØØØØØØØØØØØØØØØ
"):: SUBEND
1150 SUB HUT
1160 CALL CHAR(124,"Ø305Ø7ØD
Ø6Ø3Ø1Ø11F3F7C6764676467CØAØ
E1B363C3878EFCFC38EØ2ØEØ2ØEØ
")
1170 CALL CHAR(120,"8Ø7ØF8DC
ØCØ4Ø2ØØØØØØØØØØØØØØØØØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØØØØ
")
1180 CALL CHAR(116,"76776707
Ø7Ø6Ø6Ø6Ø6ØCØCØCØC1C3CØØ6ØEØ
EØEØE0606060603030303038 3C00
"):: SUBEND
1190 SUB HAMPEL
1200 CALL CHAR(124,"Ø305Ø7CE
C6C3E1713F3F1CØ704Ø704Ø7CØAØ
EØ706ØCØ8Ø8ØF8FC3EE626E626E6
")
1210 CALL CHAR(120,"Ø303ØFØØ
CØEØØØØØØØØØØØØØØØØØØØØCØCØ
FØØØØØØØØØØØØØØØØØØØØØØØØ
")
1220 CALL CHAR(116,"Ø6Ø7Ø7Ø7
Ø7Ø6Ø6Ø6Ø6ØEØCØCØC1C3CØØ6EEE
E6EØE0606070783C1EØE060E1EØØ
")
1230 CALL CHAR(112,"ØØØCØCØC
ØCØØØØØØØØØØØØØØØØØØØØ183C
1EØEØØØØØØØØØØØØØØØØØØØØ
"):: SUBEND
1240 SUB SPRUNG
1250 CALL CHAR(124,"Ø305Ø7ØC
Ø7Ø3Ø1Ø11FFFFCØ704Ø704Ø7CØAØ
EØ3ØEØCØ8Ø8ØF8FF3FEØ2ØEØ2ØEØ
")
1260 CALL CHAR(120,"Ø6Ø7Ø7Ø7
8EFEFC78ØØØØØØØØØØØØØØ6ØEØ
EØEØ717F3F1EØØØØØØØØØØØØØØ
")
1270 CALL CHAR(116,"ØØØØØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØ2Ø3
Ø3ØØØØØØØØØØØØØØØØØØ3Ø3Ø3Ø2
")
1280 CALL CHAR(112,"4ØCØCØØØ
ØØØØØØØØØØØØØØCØCØCØ4ØØØØØ
ØØØØØØØØØØØØØØØØØØØØØØØØØØ
```

```
")
1290 CALL CHAR(108,"00000000
80F0F878000000000000000000000
0000010F1F1E0000000000000000
"):: SUBEND
1300 SUB ROLLE
1310 CALL CHAR(124,"0305070E
0603011F3C7F66270F0C0000C0A0
E07060C080F83CFE66E4F0300000
")
1320 CALL CHAR(120,"000000C0
C0001C3C00000000000000000000
0003030038 3C0000000000000000
")
1330 CALL CHAR(116,"00000010
3C7C7E3C303E77EDFBF61D070000
00000000000000000020E0E0C0C0
")
1340 CALL CHAR(112,"403C1E0F
0301000000000000000C0E00000
00000000810303030000020606 06
")
1350 CALL CHAR(108,"00000000
00000000000000000000000000000
000000000000041C381800003838
")
1360 CALL CHAR(104,"00000000
00000000000000040703030300000
00083C3E7E3C0C7CEEB7DF6FB8E0
")
1370 CALL CHAR(100,"00000000
000081C0C0C00000406060600 23C
78F0C08000000000000000000307
")
1380 CALL CHAR(96,"000000000
0000000203 81C1800001C1C00000
00000000000000000000000000"
):: SUBEND
1390 SUB MANN1
1400 CALL HCHAR(12,21,141)::
 CALL HCHAR(12,22,143):: CAL
L HCHAR(13,21,128):: CALL HC
HAR(13,22,130):: CALL HCHAR(
14,21,129)
1410 CALL HCHAR(14,22,131)::
 CALL HCHAR(15,21,132):: CAL
L HCHAR(15,22,134):: CALL HC
HAR(16,21,133):: CALL HCHAR(
16,22,135):: SUBEND
1420 SUB ZAEHL(ZAHL)
1430 ZAHL=ZAHL-1 :: DISPLAY
AT(24,26)BEEP:ZAHL :: SUBEND
1440 SUB MANN2
1450 CALL HCHAR(12,8,141)::
CALL HCHAR(12,9,143):: CALL
HCHAR(13,8,128):: CALL HCHAR
(13,9,130):: CALL HCHAR(14,8
,129)
1460 CALL HCHAR(14,9,131)::
CALL HCHAR(15,8,132):: CALL
HCHAR(15,9,134):: CALL HCHAR
(16,8,133):: CALL HCHAR(16,9
,135):: SUBEND
1470 SUB CLEAN
1480 CALL VCHAR(12,8,32,5)::
 CALL VCHAR(12,9,32,5):: CAL
L VCHAR(12,21,32,5):: CALL V
CHAR(12,22,32,5):: SUBEND
1490 SUB TIME
1500 RESTORE 1510 :: FOR I=9
2 TO 95 :: READ A$ :: CALL C
HAR(I,A$):: NEXT I
1510 DATA 00EA4B4A4A4A0000,0
02E68AEA82E0000,001D1515151D
0000,005C484848C80000
1520 DISPLAY AT(24,24)BEEP:"
\]^_" :: FOR Q=1 TO 700 :: N
EXT Q :: CALL HCHAR(24,26,32
,4):: SUBEND
1530 SUB DEMO
1540 RESTORE 1550 :: FOR I=9
2 TO 94 :: READ A$ :: CALL C
HAR(I,A$):: NEXT I
1550 DATA 00C7A4949794A4C7,0
0A236362A2A22A2,00F090909090
90F0
1560 DISPLAY AT(19,13)BEEP:"
\]^" :: SUBEND
1570 SUB WRONG
1580 RESTORE 1590 :: FOR I=9
2 TO 94 :: READ A$ :: CALL C
HAR(I,A$):: NEXT I
1590 DATA 00E1928AF38A92E2,0
0CE2928E828292E,000C0C8C8C80
0C0C
1600 DISPLAY AT(20,11)BEEP:"
\]^" :: FOR Q=1 TO 500 :: NE
XT Q :: CALL HCHAR(20,13,32,
3):: SUBEND
1610 SUB GOOD
1620 RESTORE 1630 :: FOR I=9
2 TO 94 :: READ A$ :: CALL C
HAR(I,A$):: NEXT I
1630 DATA 00F784B49494F700,0
0BDA5A5A5A5BD00,00834323234 0
8300
1640 DISPLAY AT(20,11):"\]^"
 :: FOR Q=1 TO 300 :: NEXT Q
 :: CALL HCHAR(20,13,32,3)::
 SUBEND
1650 SUB PUNKTE(MERK,ZAHL)
1660 IF MERK=0 THEN PU=0 ::
SUBEXIT
1670 PU=PU+(MERK*ZAHL*5):: D
ISPLAY AT(24,16):PU
1680 IF HI<PU THEN CALL HISC
 :: HI=PU :: DISPLAY AT(24,4
)SIZE(7):HI
1690 SUBEND
1700 SUB HISC
1710 RESTORE 1720 :: FOR I=9
2 TO 95 :: READ A$ :: CALL C
HAR(I,A$):: NEXT I
1720 DATA 00ABAAEBA8AB0000,0
0BB22A2A2BB0000,00BBAABBB2AB
0000,0084029F02840000
```

```
1720 DISPLAY AT(24,12)SIZE(4
)BEEP:"\]^_" :: FOR I=1 TO 5
00 :: NEXT I :: CALL HCHAR(2
4,14,32,4):: SUBEND
1740 SUB FARB
1750 IF FA=0 THEN 1770
1760 FA=0 :: CALL COLOR(3,2,
1,4,2,1,5,2,1,6,2,1):: SUBEX
IT
1770 FA=1 :: CALL COLOR(3,5,
1,4,5,1,5,5,1,6,5,1):: SUBEN
D
```

BÖRSE

TI 99/4A, Ext. Basic, Siemens-Rekorder mit Kabel, Video-Chess, ca. 30 Zeitschr. 5 Bücher, 2 Joysticks, Progr., Superpr. Kompl. f. 450,– DM Tel: 09261/20790, nachmitt.

Zu verk. TI 99/4A m. Joysticks. Module: Datenverw. u. Analyse, Statistik, Personal-Generator, Kass: Haushaltsprogr. Div. Literatur, VB: 400,– DM. Tel: 0461/51509

Verk. TI-Module: Soccer 25,–, TI-Invader 30,–, Alpiner 30,–, Mash 40,–, Speech Editor 50,–. Suche: Editor/Assembler. Peter Glöde, Dorfstr. 43, 2362 Wahlstedt

---

Suche dringend einen Sprach-Synthes. f. TI 99/4A. Mögl. m. Anleit.! Bezahle für das Gerät in 1A-Zustand + Versand u. NN max. 90,– DM. Bitte melden bei: Andreas Scholz, Londoner Ring 6, 6700 Ludwigshafen, Tel: 0621/666424

KABOOMMM
Verk. Module! Z.B. Pole-Position, Congo-Bongo, Fathom 40,–! Car Wars, Wumpus u.s.w. 20,–! TE-2 Modul 60,– u.s.w.!!! Mehr Info über weitere Module (24 insges.) gibt es bei: Manfred Lipowski, In der Wanne 165, 462 Castrop-Rauxel 4, Tel: 02305/72237. Auch Tausch geg. gleichwert. Module!

Bernstein-Monitor m. Kabel f. TI 99/4A f. 300,– DM zu verk. Johann Schmitz, Pf. 900 771, 5000 Köln 90

Verk.: TI99/4A + Kass.-Kabel + orig. X-Basic + TI-Joysticks + Spiele + Bücher + 3 Spielmodule zu TI (Parsec + Schach + Munch Man) + 9 TI-Revue + 1 TI-Special. Nur zusammen, Preis: 520,– DM. Tel: 05064/6177

TI 99/4A: 2 Konsolen, Ex-Basic, P-Box (defekt), RS 232-Karte, Disk-Drive, 32 K-Karte, Editor-Assembler, TI-Writer, Joystick Brother CE-60 m. Interface, Multiplan u. 7 Spielmod. auch einzeln zu verkaufen. Tel: 069/504575

Suche ext. 90 K-Laufw. f. orig. TI-Contr. (mögl. m. eig. Stromvers.) sowie Sprachsynt. Angebote an: Hauke Hölterling, Carl-Schutz-Str. 15, 6070 Langen. Tel: 06103/72518

Tips & Tricks für TI 99/4A, Teile I + II, jeweils 1 - 3, 5 KB (Ex-Basic). Progr. auf Kass. geg. Einsendung von 10,– DM. Peter Hielscher, Am Wall 22, 4401 Saerbeck

---

TI 99/4A Konsole m. eingeb. 32 K-CMOS Erw. zu verk. Rechner ist z.B. in Assembl. 2 x so schnell wie mit normalen 32 K, da alle 16 Datenleitungen direkt am TMS 9900 angeschl. sind. Nähere Info: Tel: 06257/83247

TI 99/4A Box Floppy m. Karte Ex-Modul RS 232 + Karte. 32 K Multiplan TE II Diskmanager, Drucker GP-100 Zenith-Monit. Joystick, Dateiverwaltung, 3 D Welt. Alle Handbücher. Div. Kabel. Preis VB. Alles auch einzeln. Hoffmann, Tel: 07022/46542, ab 12 Uhr

---

Verk. TI 99/4A mit eingeb. 32K-Byte Erw. + Ex-Basic Modul f. 400,– DM, Tel: 06201/75170, Kohout

---

Verk. TI 99/4A Konsole, 3 J. alt + Software + Rec. Kabel + Bücher, VB 200,– DM. Tel: 02761/61749

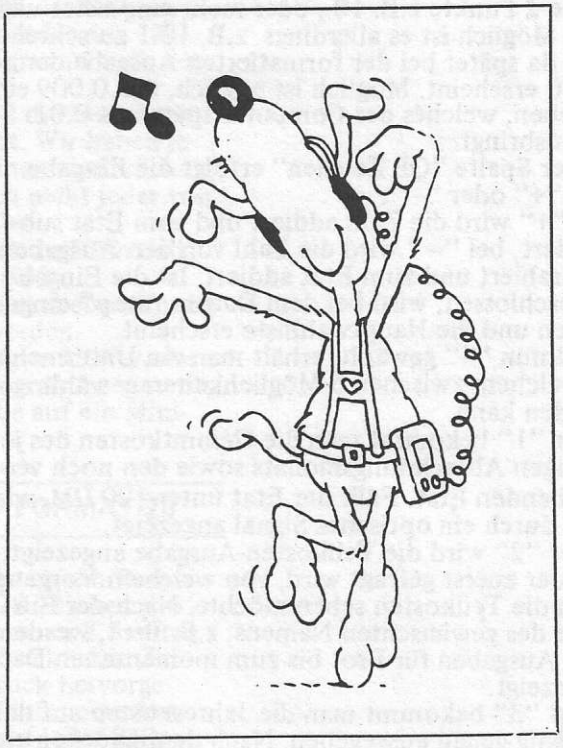
Suche: ROM + GROM d. RS232 a. Cass. gespeich.! H.P. Thelen, Sebastianstr. 165, 5300 Bonn 1, Tel: 0228/628907

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *      CYCLONS         *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *  Matthias Renken     *
16 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
24 ! *  Joystick 1 u. 2     *
25 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     5319 Bytes       *
28 ! *                      *
29 ! ************************
30 !
100 RANDOMIZE :: RESTORE
110 CALL CLEAR :: CALL SCREE
N(2):: CALL COLOR(9,4,2,5,4,
2,6,4,2,7,4,2,3,4,2)
120 CALL COLOR(4,4,2,8,4,2)
130 ! SONDERZEICHEN
140 CALL CHAR(63,"EA4E4A0000
000000",64,"3C4299A1A199423C
")
150 CALL CHAR(49,"0004040404
040404",50,"003C04043C20203C
")
160 CALL CHAR(51,"0038040438
040438",52,"0020202241E040404
")
170 CALL CHAR(53,"003C20203C
04043C")
180 CALL CHAR(56,"003C242418
24243C",57,"003C24243C04043C
")
190 CALL CHAR(104,"00000000F
FFF0000",105,"FFFFFFFFFFFF00
00")
200 !UFO
210 CALL CHAR(124,"0000000000
0000003070F3F7FF67F1F0000000
00000000000C0E0F8FCDEFCF000"
)
220 !GESCHOSS
230 CALL CHAR(120,"0000000000
0000010100000000000000000000
00000000008080000000000000000"
)
240 !UFO BEI SCHUSS
250 CALL CHAR(116,"0000000000
000000304083040089601F0000000
00000000000804020180422CF000"
)
260 ! EXPLOSION
270 CALL CHAR(112,"0002802000
40007550301052400044200001000
84000C8C0C082B040400C4000000"
)
280 ! HINTERGRUNDSTERNE
290 CALL CHAR(98,"0000000010
000000",99,"0000001038100000
```

```
")
300 CALL MAGNIFY(3)
310 GOSUB 1180
320 FOR F=1 TO 5 :: READ ZE,
SP,TE$
330 FOR E=1 TO LEN(TE$):: CA
LL HCHAR(ZE,SP+E,ASC(SEG$(TE
$,E,1)))
340 CALL SOUND(-10,INT(RND*5
000)+200,0)
350 NEXT E
360 NEXT F
370 FOR W=1 TO 25 :: S=INT(R
ND*13)+3 :: CALL COLOR(3,S,2
,4,S,2,5,S,2,6,S,2,7,S,2,8,S
,2):: NEXT W
380 CALL KEY(1,KEY,ST):: IF
ST<>0 THEN 400
390 CALL KEY(2,KEY,ST):: IF
ST=0 THEN 380
400 CALL CLEAR
410 CALL COLOR(10,10,2,3,2,1
0)
420 GOSUB 1210
430 GOSUB 1180
440 CALL HCHAR(20,5,104,3)::
 CALL HCHAR(20,23,104,3):: C
ALL HCHAR(21,5,105,3):: CALL
 HCHAR(21,23,105,3)
450 CALL HCHAR(21,6,ZAHLE+48
):: CALL HCHAR(21,24,ZAHLZ+4
8)
460 RX,RY,RX2,RY2,DELX,DELY,
DELX2,DELY2=0
470 CALL SPRITE(#1,124,12,14
1,37)
480 CALL SPRITE(#2,124,5,141
,181)
490 CALL JOYST(1,X,Y):: IF X
=0 AND Y=0 THEN DELY,DELX=0
:: GOTO 610
500 IF Y=4 THEN DELX=-3 :: G
OTO 550
510 IF Y=-4 THEN DELX=3 :: G
OTO 550
520 IF X=4 THEN DELY=3 :: GO
TO 550
530 IF X=-4 THEN DELY=-3 ::
GOTO 550
540 DELX,DELY=0
550 RX=RX+DELX :: IF RX>=MA
THEN RX=MA ELSE IF RX<=-MA T
HEN RX=-MA
560 RY=RY+DELY :: IF RY>=MA
THEN RY=MA ELSE IF RY<=-MA T
HEN RY=-MA
570 CALL KEY(1,KEY,ST):: IF
KEY=18 THEN GOSUB 740
580 CALL MOTION(#1,RX,RY)
590 CALL COINC(#1,#2,COIC,I)
:: IF I=0 THEN 610
600 GOTO 940
610 CALL JOYST(2,X2,Y2):: IF
 X2=0 AND Y2=0 THEN DELX2,DE
LY2=0 :: GOTO 710
620 IF Y2=4 THEN DELX2=-3 ::
 GOTO 670
630 IF Y2=-4 THEN DELX2=3 ::
 GOTO 670
640 IF X2=4 THEN DELY2=3^::
GOTO 670
650 IF X2=-4 THEN DELY2=-3 :
: GOTO 670
660 DELX2,DELY2=0
670 RX2=RX2+DELX2 :: IF RX2>
=MA THEN RX2=MA ELSE IF RX2<
=-MA THEN RX2=-MA
680 RY2=RY2+DELY2 :: IF RY2>
=MA THEN RY2=MA ELSE IF RY2<
=-MA THEN RY2=-MA
690 CALL KEY(2,KEY,ST):: IF
KEY=18 THEN GOSUB 840
700 CALL MOTION(#2,RX2,RY2)
710 CALL COINC(#1,#2,COIC,I)
:: IF I=0 THEN 730
720 GOTO 940
730 GOTO 490
740 REM ***SCHUSS***
750 CALL POSITION(#1,Q,W)::
CALL PATTERN(#1,116)
760 CALL SPRITE(#3,120,16,Q,
W,SDG*DELX+RX,SDG*DELY+RY)
770 FOR F=900 TO 1000 STEP 2
0
780 CALL SOUND(-10,F,0):: CA
LL COINC(#3,#2,COIS,A):: IF
A<>0 THEN 800
790 NEXT F :: CALL PATTERN(#
1,124):: CALL DELSPRITE(#3):
: RETURN
800 CALL DELSPRITE(#3):: CAL
L MOTION(#2,0,0):: CALL PATT
ERN(#2,112):: CALL SOUND(100
0,-7,0)
810 FOR T=1 TO 700 :: NEXT T
 :: CALL DELSPRITE(ALL)
820 ZAHLZ=ZAHLZ-1 :: IF ZAHL
Z<=0 THEN 1020
830 CALL HCHAR(21,24,ZAHLZ+4
8):: GOTO 460
840 REM ***schuss 2***
850 CALL POSITION(#2,Q,W)::
CALL PATTERN(#2,116)
860 CALL SPRITE(#4,120,16,Q,
W,SDG*DELX2+RX2,SDG*DELY2+RY
2)
870 FOR F=900 TO 1000 STEP 2
0
880 CALL SOUND(-10,F,0):: CA
LL COINC(#4,#1,COIS,A):: IF
A<>0 THEN 900
890 NEXT F :: CALL PATTERN(#
2,124):: CALL DELSPRITE(#4):
: RETURN
900 CALL DELSPRITE(#4):: CAL
L MOTION(#1,0,0):: CALL PATT
```

```
ERN(#1,112):: CALL SOUND(100
0,-7,0)
910 FOR T=1 TO 700 :: NEXT T
 :: CALL DELSPRITE(ALL)
920 ZAHLE=ZAHLE-1 :: IF ZAHL
E<=0 THEN 1020
930 CALL HCHAR(21,6,ZAHLE+48
):: GOTO 460
940 REM kollision
950 CALL MOTION(#1,0,0,#2,0,
0)
960 CALL PATTERN(#1,112):: C
ALL PATTERN(#2,112):: CALL S
OUND(1000,-7,0)
970 FOR T=1 TO 700 :: NEXT T
 :: CALL DELSPRITE(ALL)
980 ZAHLE=ZAHLE-1 :: ZAHLZ=Z
AHLZ-1
990 IF ZAHLE=0 OR ZAHLZ=0 TH
EN 1020
1000 CALL HCHAR(21,6,ZAHLE+4
8):: CALL HCHAR(21,24,ZAHLZ+
48)
1010 GOTO 460
1020 CALL CLEAR
1030 FOR S=1 TO 3 :: READ TE
$
1040 FOR Y=1 TO LEN(TE$):: C
ALL HCHAR(6+S*2,Y+9,ASC(SEG$
(TE$,Y,1))):: NEXT Y
1050 NEXT S
1060 IF ZAHLE=0 AND ZAHLZ<>0
 THEN DISPLAY AT(14,8):"PLAY
ER 2" :: GOTO 1090
1070 IF ZAHLE<>0 AND ZAHLZ=0
 THEN DISPLAY AT(14,8):"PLAY
ER 1" :: GOTO 1090
1080 DISPLAY AT(14,8):"NOBOD
Y !!!"
1090 CALL KEY(1,KEY,ST):: IF
 ST<>0 THEN 1110
1100 CALL KEY(2,KEY,ST):: IF
 ST=0 THEN 1090
1110 CALL CLEAR :: ZAHLE,ZAH
LZ=3 :: RESTORE 1170
1120 GOTO 400
1130 CALL CLEAR :: END
1140 DATA 5,11,"MARE-SOFT?",
7,11,"PROUDLY"
1150 DATA 9,11,"PRESENTS",12
,9,">> CYCLONS <<"
1160 DATA 20,3,"@1985 BY MAT
THIAS RENKEN"
1170 DATA "THIS   GAME","IS
    OVER","WINNER   IS"
1180 FOR E=1 TO 20 :: CALL H
CHAR(INT(RND*20)+1,INT(RND*3
0)+1,98):: NEXT E
1190 FOR E=1 TO 10 :: CALL H
CHAR(INT(RND*20)+1,INT(RND*2
9)+1,99):: NEXT E
1200 RETURN
1210 REM MENUE
1220 DISPLAY AT(2,2):"PLEASE
 CHOOSE DIFFICULTY" :: DISPL
AY AT(4,2):"LEVEL :"
1230 DISPLAY AT(7,2):"1 = IN
NOCENT"
1240 DISPLAY AT(9,2):"2 = AD
VANCED"
1250 DISPLAY AT(11,2):"3 = E
XPERT"
1260 DISPLAY AT(13,2):"4 = C
RAZY"
1270 DISPLAY AT(18,2):"PRESS
 CHOOSEN KEY"
1280 CALL KEY(0,L,J):: IF J=
0 THEN 1280
1290 IF L<49 OR L>52 THEN CA
LL SOUND(200,500,2):: GOTO 1
280
1300 ON L-48 GOTO 1310,1330,
1350,1370
1310 SDG=8 :: MA=12 :: COIS=
10 :: COIC=10
1320 CALL CLEAR :: GOTO 1390
1330 SDG=10 :: MA=15 :: COIS
=7 :: COIC=13
1340 CALL CLEAR :: GOTO 1390
1350 SDG=12 :: MA=25 :: COIS
=5 :: COIC=14
1360 CALL CLEAR :: GOTO 1390
1370 SDG=14 :: MA=32 :: COIS
=3 :: COIC=15
1380 CALL CLEAR :: GOTO 1390
1390 CALL CLEAR
1400 DISPLAY AT(10,2):"HOW M
ANY SHIPS <1-5>:"
1410 ACCEPT AT(12,2):HIL ::
IF HIL<1 OR HIL>5 THEN 1410
1420 ZAHLE,ZAHLZ=HIL :: CALL
 CLEAR :: RETURN
```



HAUSHALTS-RECHNUNG

Nach dem Start bekommt man nach kurzer Zeit eine Hauptwahlliste zu sehen.
Begibt man sich über "3" in die Datenein- und -ausgabe, kommt als nächstes die Frage nach dem Datum des Datenstandes. Danach erfolgt die Eingabe des Etats, den man für den Abrechnungszeitraum zur Verfügung hat, z.B. 1200 DM. Sollte während diesem Zeitraum das Geld knapp werden, so besteht die Möglichkeit, den Etat aufzustocken, z.B. um 200 DM, was den Gesamtetat auf 1400 DM anhebt.
Sodann erfolgt die Eingabe der eigentlichen Daten:
Z.B.:

Datum	Firma	Betrag	OP
01.01	Müller	10.20	+
	(oder bei Sachbezogen Brot etc.)		Operation bei Ausgaben +

Es besteht die Möglichkeit, 108 Datenreihen untereinander einzugeben. Danach wird darauf hingewiesen, daß keine Eingabe mehr möglich ist.
Alle Eingaben im Programm werden auf ihre Plausibilität und Größe überprüft.
Bei der Daten-Eingabe muß bei einstelligen Tages- und Monatszahlen jeweils zweistellig eingegeben werden z.B. 01.01.
Der Firmenname ist auf seine Länge begrenzt.
Beim Betrag werden nur Werte, die kleiner oder gleich 999.99 sind, zugelassen. Es können auch keine 2 Punkte z.B. 10.. oder mehr eingesetzt werden. Möglich ist es allerdings, z.B. 10.1 zu schreiben, da später bei der formatierten Ausgabe dann 10.10 erscheint. Möglich ist es auch, z.B.0.009 einzugeben, welches der Computer später als 0.01 herausbringt.
In der Spalte "OP Zeichen" erfolgt die Eingabe von "+" oder "–".
Bei "+" wird die Zahl addiert und vom Etat subtrahiert, bei "–" wird die Zahl von den Ausgaben subtrahiert und zum Etat addiert. Ist die Eingabe abgeschlossen, wird bei dem Datum "stop" eingetragen und die Hauptwahlliste erscheint.
Wird nun "4" gewählt, erhält man ein Untermenü, in welchem zwischen 5 Möglichkeiten gewählt werden kann.
Über "1" bekommt man die Gesamtkosten des jeweiligen Abrechnungsmonats sowie den noch verbleibenden Etat. Fällt der Etat unter 100 DM, wird dies durch ein optisches Signal angezeigt.
Über "2" wird die Teilkosten-Ausgabe angezeigt, bei der zuerst gefragt wird, von welchem Vorgang man die Teilkosten sehen möchte. Nach der Eingabe des gewünschten Namens, z.B. Brot, werden die Ausgaben für Brot bis zum momentanen Datum angezeigt.
Über "3" bekommt man die Jahreskosten auf den Pfennig genau ausgegeben. Nach dem Drücken einer x-beliebigen Taste gelangt man wieder ins Untermenü.
Wird nun "4" gewählt, erscheint die Jahresgrafik. Die Genauigkeit ist hier allerdings nicht mehr so hoch wie bei der Jahreskosten-Ausgabe, aber zur Tendenzanzeige genügt dies vollauf. So werden Werte, die unter 50 DM liegen, nicht angezeigt und solche die über 1950 DM liegen, mit 2000 DM angezeigt. Um die Grafik wieder zu verlassen, wird eine x-beliebige Taste gedrückt. Danach ist man im Untermenü und nach der Eingabe von "5" im Hauptmenü.
Drücken wir nun "2", werden die Eingaben und Berechnungen abgespeichert.
Werden am nächsten Tag erneut Eintragungen vorgenommen, liest man über "1" erst einmal die alten Daten ein.
Man muß nun entscheiden, ob die Monats- oder Jahres-Daten gelöscht werden sollen. Dieses Löschen ist an sich kein richtiges Löschen, denn die Daten, die man bei der Frage nach dem Löschen mit "ja" beantwortet hat, werden nur nicht eingelesen und somit beim späteren Speichern überschrieben.
Die Jahresdaten werden sowieso : nur am Ende des Jahres und die Monatsdaten am Ende des Abrechnungszeitraums gelöscht.
Nachdem nun die gewünschten Daten eingelesen sind, geht über "3" der ganze Ablauf wieder von Anfang los.
Die schon eingegebenen Daten werden, wie bereits erwähnt, formatiert ausgegeben. Am Ende dieser Ausgabe erscheint der Cursor und nun kann weiter eingegeben werden. Ist eine Seite vollständig ausgegeben, drückt man eine x-beliebige Taste und die zweite Seite wird ausgegeben.
Bei der Eingabe wird nach jeder vollen Seite der Bildschirm gelöscht und der Cursor steht wieder links oben und wartet auf neue Eingaben.
Die schon erwähnten 108 Datenzeilen, entsprechen 4 1/2 Seiten.
Diese Programmbeschreibung liest sich bestimmt sehr trocken, und man begreift soviel wie bei mancher Betriebsanleitung. Vielleicht liegt es auch am Verfasser der Erklärung. Es ist deswegen am besten, man probiert alles selbst aus, dann bereitet es auch (hoffentlich) mehr Freude.

Ansonsten bleibt dem Anwender jede Möglichkeit offen, mit dem Programm zu experimentieren und vielleicht hat so mancher noch eine gute Idee, die er anbringen kann.
Weiterhin dürfte es keine Schwierigkeiten bereiten, das Programm für die Anwendung ohne 32 K-Erweiterung und Cassettenrecorder umzuschreiben oder die Kosten und Übersichten über einen Drukker auszugeben. Die vorhandene Datei zu erweitern, wird wohl auch keine Probleme aufwerfen.

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *  Haushaltsfuehrung   *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *    Reiner Kurtz      *
16 ! *                      *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *  32K-Erweiterung     *
23 ! *  Diskettenstation    *
24 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     13431 Bytes      *
28 ! *                      *
29 ! ************************
30 !
170 REM *  DIMENS. FELDER   *
180 REM ************************
190 !@P+
200 CALL CLEAR
210 OPTION BASE 1
220 DIM FIR$(108),DA$(108),B
ET$(108),OP$(108),MON(12),MO
(12)
230 GOTO 260 :: CALL CLEAR :
: CALL SCREEN :: CALL CHAR :
: CALL MAGNIFY :: CALL SPRIT
E :: CALL GCHAR :: CALL DELS
PRITE :: CALL KEY :: CALL HC
HAR :: CALL VCHAR :: CALL CO
LOR :: Z :: L6
240 SS :: SI :: SIC :: S ::
K :: I :: DS :: F :: P :: DA
T$ :: D1$ :: DAT1$ :: A$ ::
B$ :: B :: C$ :: BA$ :: BA :
: L :: L1 :: L2 :: L3 :: L4
:: L5 :: L$ :: L1$ :: L2$ ::
 L3$ :: L4$ :: L5$ :: L6$ ::
 ML$ :: JL$ :: G :: GK :: BK
 :: FR$ :: TK :: T :: D$ ::
BIL
250 !@P-
260 CALL SCREEN(3)
270 DISPLAY AT(12,7):"HAUSHA
LTSFUEHRUNG"
280 CALL CHAR(104,"3F24243F2
A202A202A202AFF808380FFFC949
4FCA414A414A404A4FF01C101FF"
)
290 CALL MAGNIFY(4)
300 CALL SPRITE(#2,104,11,12
0,116)
310 CALL CHAR(64,"3C4299A1A1
99423C")
320 DISPLAY AT(24,1):"@ RK S
OFTWARE"
330 CALL GCHAR(24,5,SI)
340 CALL GCHAR(24,6,SIC)
350 FOR BILD=1 TO 1000 :: NE
XT BILD
360 REM ********************
370 REM *    HAUPTMENUE    *
380 REM ********************
390 CALL DELSPRITE(ALL)
400 DISPLAY AT(2,7)ERASE ALL
:"HAUSHALTSFUEHRUNG"
410 DISPLAY AT(8,3):"1=> DAT
EN EINLESEN"
420 DISPLAY AT(11,3):"2=> DA
TEN SPEICHERN"
430 DISPLAY AT(14,3):"3=> DA
TEN EIN-UND AUSGABE"
440 DISPLAY AT(17,3):"4=> KO
STENAUSGABE/GRAPHIK"
450 DISPLAY AT(20,3):"5=> EN
DE"
460 DISPLAY AT(23,3):"BITTE
WAEHLEN SIE"
470 SS=SI+SIC
480 CALL KEY(0,K,S)
490 IF S=0 THEN 480
500 IF K<49 THEN 480
510 IF K>53 THEN 480
520 ON K-48 GOTO 2230,1960,5
70,2600,530
530 END
540 REM ********************
550 REM *  DATUMS EINGABE  *
560 REM ********************
570 DISPLAY AT(4,3)ERASE ALL
:"DATENSTAND VOM"
580 DISPLAY AT(6,3):DAT$
590 DISPLAY AT(9,3):"MOECHTE
N SIE EIN NEUES"
600 DISPLAY AT(11,3):"DATUM
EINSETZEN ?(J/N)"
610 ACCEPT AT(11,26)SIZE(8)B
EEP VALIDATE("J","N"):D1$
620 IF D1$="" THEN 610
630 IF D1$="N" THEN 870
640 DISPLAY AT(14,3):"DATUM"
650 ACCEPT AT(14,10)SIZE(8)B
EEP VALIDATE(DIGIT,"."):DAT1
$
660 L2$=SEG$(DAT1$,1,2)
670 L3$=SEG$(DAT1$,4,2)
680 IF L2$>"31" OR L2$<="00"
 OR L3$>"12" OR L3$<="00" TH
EN 650
690 L4=LEN(DAT1$)
700 IF L4<8 THEN 650
710 L2=POS(DAT1$,".",1)
720 IF L2=3 THEN 730 ELSE 65
0
730 L3=POS(DAT1$,".",4)
740 IF L3=6 THEN 750 ELSE 65
0
750 L4$=SEG$(DAT1$,7,1)
760 IF L4$="." THEN 650 ELSE
 770
770 L5$=SEG$(DAT1$,8,1)
780 IF L5$="." THEN 650 ELSE
 790
```

```
790 IF DAT1$="" THEN DAT$=DA
T$ ELSE DAT$=DAT1$
800 DISPLAY AT(21,3):"BITTE
DRUECKEN SIE EINE"
810 DISPLAY AT(23,3):"TASTE"
820 CALL KEY(0,K,S)
830 IF S=0 THEN 820
840 REM ********************
850 REM *   ETAT EINGABE   *
860 REM ********************
870 DISPLAY AT(4,3)ERASE ALL
:"MOECHTEN SIE EINEN NEUEN"
880 DISPLAY AT(6,3):"ETAT EI
NSETZEN ?(J/N)"
890 ACCEPT AT(6,25)SIZE(1)VA
LIDATE("J","N")BEEP:A$
900 IF A$="" THEN 890
910 IF A$="N" THEN 1060
920 DISPLAY AT(8,3):"HOEHE D
ES BETRAGES ?"
930 DISPLAY AT(10,8):"DM"
940 ACCEPT AT(10,3)SIZE(4)VA
LIDATE(DIGIT,".")BEEP:B$
950 IF B$="" THEN B=B :: GOT
O 1060
960 L5=0
970 FOR Z=1 TO 4
980 L6$=SEG$(B$,Z,1)
990 IF L6$="." THEN L5=L5+1
1000 NEXT Z
1010 IF L5>1 THEN 940
1020 B=VAL(B$)
1030 REM ********************
1040 REM * ETAT AUFSTOCKEN *
1050 REM ********************
1060 DISPLAY AT(13,3):"MOECH
TEN SIE DEN ETAT"
1070 DISPLAY AT(15,3):"AUFST
OCKEN ?(J,N)"
1080 ACCEPT AT(15,21)SIZE(1)
VALIDATE("J","N")BEEP:C$
1090 IF C$="" THEN 1080
1100 IF C$="N" THEN 1220
1110 DISPLAY AT(17,3):"HOEHE
 DES BETRAGES ?"
1120 DISPLAY AT(19,8):"DM"
1130 ACCEPT AT(19,3)SIZE(4)V
ALIDATE(DIGIT,".")BEEP:BA$
1140 IF BA$="" THEN BA=0 ::
GOTO 1220
1150 L5=0
1160 FOR Z=1 TO 4
1170 L6$=SEG$(BA$,Z,1)
1180 IF L6$="." THEN L5=L5+1
1190 NEXT Z
1200 IF L5>1 THEN 1130
1210 BA=VAL(BA$)
1220 B=B+BA
1230 IF SS=157 THEN 1240 ELS
E 1250
1240 BA=0
1250 DISPLAY AT(22,3):"BITTE
 DRUECKEN SIE EINE"
1260 DISPLAY AT(24,3):"TASTE
"
1270 CALL KEY(0,K,S)
1280 IF S=0 THEN 1270
1290 REM ********************
1300 REM *  DATEN AUSGABE  *
1310 REM ********************
1320 DISPLAY AT(1,2)ERASE AL
L:"DATUM"
1330 DISPLAY AT(1,10):"FIRMA
"
1340 DISPLAY AT(1,18):"BETRA
G"
1350 DISPLAY AT(1,27):"OP"
1360 FOR I=1 TO DS
1370 F=I+2
1380 IF I=23 OR I=47 OR I=71
 OR I=95 OR I=108 THEN GOSUB
 2530
1390 IF I=108 THEN 1880
1400 DISPLAY AT(F,2):DA$(I)
1410 DISPLAY AT(F,10):FIR$(I
)
1420 IF BET$(I)="" THEN 1460
1430 BET=VAL(BET$(I))
1440 DISPLAY AT(F,18):USING
"###.##":BET
1450 DISPLAY AT(F,27):OP$(I)
1460 NEXT I
1470 REM ********************
1480 REM *  DATEN EINGABE  *
1490 REM ********************
1500 FOR DS=DS TO 107
1510 IF DS=0 THEN DS=1
1520 F=DS+2
1530 ACCEPT AT(F,2)SIZE(5)BE
EP VALIDATE(DIGIT,".","STOP"
);DA$(DS)
1540 IF DA$(DS)="" THEN 1530
1550 IF DA$(DS)="STOP" THEN
1560 ELSE 1610
1560 BET$(DS)=""
1570 DA$(DS)=""
1580 FIR$(DS)=""
1590 OP$(DS)=""
1600 GOTO 400
1610 L=POS(DA$(DS),".",1)
1620 IF L<>3 THEN 1530
1630 L1=LEN(DA$(DS))
1640 IF L1<5 THEN 1530
1650 L$=SEG$(DA$(DS),1,2)
1660 IF L$>"31" OR L$<="00"
THEN 1530
1670 L1$=SEG$(DA$(DS),4,2)
1680 IF L1$>"12" OR L1$<="00
" OR L1$="1." THEN 1530
1690 ACCEPT AT(F,10)SIZE(6)B
EEP VALIDATE(UALPHA):FIR$(DS
)
1700 IF FIR$(DS)="" THEN 169
0
1710 ACCEPT AT(F,18)SIZE(6)B
EEP VALIDATE(DIGIT,"."):BET$
```

LISTINGS

```
(DS)
1720 IF BET$(DS)="" THEN 171
0
1730 L5=0
1740 FOR Z=1 TO 6
1750 L6$=SEG$(BET$(DS),Z,1)
1760 IF L6$="." THEN L5=L5+1
1770 NEXT Z
1780 IF L5>1 THEN 1710
1790 L6=VAL(BET$(DS))
1800 IF L6>999.99 THEN 1710
1810 ACCEPT AT(F,27)SIZE(1)B
EEP VALIDATE("+","-"):OP$(DS
)
1820 IF OP$(DS)="" THEN 1810
1830 IF OP$(DS)="+" THEN GOS
UB 2740 ELSE IF OP$(DS)="-"
THEN GOSUB 2820
1840 GOSUB 3460
1850 IF DS=22 OR DS=46 OR DS
=70 OR DS=94 THEN CALL CLEAR
1860 NEXT DS
1870 IF DS=107 THEN FIR$(DS)
="" :: DA$(DS)="" :: BET$(DS
)"" OP$(DS):: GOTO 1880
1880 DISPLAY AT(5,4)ERASE AL
L:"SIE HABEN BEREITS 100"
1890 DISPLAY AT(7,4):"ZEILEN
 EINGEGEBEN"
1900 DISPLAY AT(9,4):"ES IST
 KEINE EINGABE MEHR"
1910 DISPLAY AT(11,4):"MOEGL
ICH"
1920 DISPLAY AT(19,4):"BITTE
 DRUECKEN SIE EINE"
1930 DISPLAY AT(21,4):"TASTE
"
1940 CALL KEY(0,K,S)
1950 IF S=0 THEN 1940 ELSE 4
00
1960 IF DS=0 THEN 1970 ELSE
2060
1970 DISPLAY AT(4,4)ERASE AL
L:"KEIN SPEICHERN MOEGLICH"
1980 DISPLAY AT(6,4):"KEINE
DATEN VORHANDEN"
1990 DISPLAY AT(19,4):"BITTE
 DRUECKEN SIE EINE"
2000 DISPLAY AT(21,4):"TASTE
"
2010 CALL KEY(0,K,S)
2020 IF S=0 THEN 2010 ELSE 4
00
2030 REM ********************
2040 REM * DATEN SPEICHERN *
2050 REM ********************
2060 DISPLAY AT(10,9)ERASE A
LL:"BITTE WARTEN"
2070 DISPLAY AT(16,3):"DATEN
 WERDEN GESPEICHERT"
2080 OPEN #1:"DSK1.MONAT",IN
TERNAL,RELATIVE,FIXED
2090 PRINT #1:DS,GK,B,DAT$
2100 FOR I=1 TO DS
2110 PRINT #1:FIR$(I),DA$(I)
,BET$(I),OP$(I)
2120 NEXT I
2130 CLOSE #1
2140 OPEN #2:"DSK1.JAHR",INT
ERNAL,RELATIVE,FIXED
2150 FOR I=1 TO 12
2160 PRINT #2:MON(I),MO(I)
2170 NEXT I
2180 CLOSE #2
2190 GOTO 400
2200 REM ********************
2210 REM * DATEN LOESCHEN  *
2220 REM ********************
2230 DISPLAY AT(4,3)ERASE AL
L:"MONATSDATEN LOESCHEN ?" :
: DISPLAY AT(6,3):"(J/N)"
2240 ACCEPT AT(6,9)SIZE(1)VA
LIDATE("J","N")BEEP:ML$
2250 IF ML$="" THEN 2240
2260 DISPLAY AT(10,3):"JAHRE
SDATEN LOESCHEN ?" :: DISPLA
Y AT(12,3):"(J/N)"
2270 ACCEPT AT(12,9)SIZE(1)V
ALIDATE("J","N")BEEP:JL$
2280 IF JL$="" THEN 2270
2290 IF ML$="J" AND JL$="J"
THEN 400
2300 REM ********************
2310 REM *  DATEN EINLESEN *
2320 REM ********************
2330 DISPLAY AT(10,9)ERASE A
LL:"BITTE WARTEN"
2340 DISPLAY AT(16,4):"DATEN
 WERDEN EINGELESEN"
2350 IF ML$="J" THEN 2420
2360 OPEN #1:"DSK1.MONAT",IN
TERNAL,RELATIVE,FIXED
2370 INPUT #1:DS,GK,B,DAT$
2380 FOR I=1 TO DS
2390 INPUT #1:FIR$(I),DA$(I)
,BET$(I),OP$(I)
2400 NEXT I
2410 CLOSE #1
2420 IF JL$="J" THEN 2490
2430 IF SS<>157 THEN 2490
2440 OPEN #2:"DSK1.JAHR",INT
ERNAL,RELATIVE,FIXED
2450 FOR I=1 TO 12
2460 INPUT #2:MON(I),MO(I)
2470 NEXT I
2480 CLOSE #2
2490 GOTO 400
2500 REM ********************
2510 REM *  UNTERPRG.TASTE *
2520 REM ********************
2530 CALL KEY(0,K,S)
2540 IF S=0 THEN 2530
2550 CALL CLEAR
2560 RETURN
2570 REM ********************
2580 REM *    UNTERMENUE   *
```

LISTINGS

```
2590 REM ********************
2600 DISPLAY AT(8,3)ERASE AL
L:"1=> GESAMTKOSTEN"
2610 DISPLAY AT(11,3):"2=> E
INZELKOSTEN"
2620 DISPLAY AT(14,3):"3=> J
AHRESKOSTEN"
2630 DISPLAY AT(17,3):"4=> J
AHRESGRAPHIK"
2640 DISPLAY AT(20,3):"5=> R
UECKSPRUNG"
2650 DISPLAY AT(23,3):"BITTE
 WAEHLEN SIE"
2660 CALL KEY(0,K,S)
2670 IF S=0 THEN 2660
2680 IF K<49 THEN 2660
2690 IF K>53 THEN 2660
2700 ON K-48 GOTO 2890,3150,
4410,4100,400
2710 REM ********************
2720 REM *      ADDITION    *
2730 REM ********************
2740 IF BET$(DS)="" THEN G=0
 :: GOTO 2770
2750 G=VAL(BET$(DS))
2760 IF SS=157 THEN 2770 ELS
E 2760
2770 GK=G+GK
2780 RETURN
2790 REM ********************
2800 REM *   SUBTRAKTION    *
2810 REM ********************
2820 IF BET$(DS)="" THEN G=0
 :: GOTO 2840
2830 G=VAL(BET$(DS))
2840 GK=GK-G
2850 RETURN
2860 REM ********************
2870 REM *   GESAMTKOSTEN   *
2880 REM ********************
2890 DISPLAY AT(3,4)ERASE AL
L:"STAND "¦DAT$
2900 CALL HCHAR(5,2,88,31)
2910 CALL HCHAR(18,2,88,31)
2920 DISPLAY AT(7,5):"ETAT"
2930 DISPLAY AT(7,18):"GES.K
OSTEN"
2940 BK=B-GK
2950 DISPLAY AT(9,4):USING "
#####.## DM":BK
2960 DISPLAY AT(9,17):USING
"#####.## DM":GK
2970 IF BK<=100 THEN DISPLAY
 AT(16,3):"ETAT MIN" :: GOTO
 2990
2980 CALL VCHAR(6,17,88,12):
: GOTO 3080
2990 CALL VCHAR(6,17,88,12)
3000 FOR BLINK=1 TO 10
3010 CALL CHAR(128,"FFFFFFFF
FFFFFFFF")
3020 CALL SPRITE(#1,128,9,89
,54)
3030 CALL MAGNIFY(2)
3040 FOR VERZ=1 TO 100 :: NE
XT VERZ
3050 CALL DELSPRITE(#1)
3060 FOR VERZ=1 TO 100 :: NE
XT VERZ
3070 NEXT BLINK
3080 DISPLAY AT(21,4):"BITTE
 DRUECKEN SIE EINE"
3090 DISPLAY AT(23,4):"TASTE
"
3100 CALL KEY(0,K,S)
3110 IF S=0 THEN 3100 ELSE 2
600
3120 REM ********************
3130 REM *    TEILKOSTEN    *
3140 REM ********************
3150 DISPLAY AT(4,4)ERASE AL
L:"VON WAS BENOETIGEN SIE"
3160 DISPLAY AT(6,4):"DIE EI
NZELKOSTEN ?"
3170 ACCEPT AT(8,4)BEEP SIZE
(6)VALIDATE(UALPHA):FR$
3180 IF FR$="" THEN 3170
3190 TK=0
3200 FOR I=1 TO DS
3210 IF FR$=FIR$(I)THEN 3330
3220 NEXT I
3230 DISPLAY AT(4,4)ERASE AL
L:"DIE KOSTEN FUER "¦FR$
3240 DISPLAY AT(6,4):"BETRAG
EN BIS ZUM "¦DAT$
3250 DISPLAY AT(8,3):USING "
 #####.## DM":TK
3260 DISPLAY AT(19,4):"BITTE
 DRUECKEN SIE EINE"
3270 DISPLAY AT(21,4):"TASTE
"
3280 CALL KEY(0,K,S)
3290 IF S=0 THEN 3280 ELSE 2
600
3300 REM ********************
3310 REM *  TEILKOST.ERMIT *
3320 REM ********************
3330 IF OP$(I)="+" THEN GOTO
 3340 ELSE 3380
3340 IF BET$(I)="" THEN T=0
:: GOTO 3360
3350 T=VAL(BET$(I))
3360 TK=T+TK
3370 GOTO 3220
3380 IF BET$(I)="" THEN T=0
:: GOTO 3400
3390 T=VAL(BET$(I))
3400 TK=TK-T
3410 IF SS<>157 THEN TK=0
3420 GOTO 3220
3430 REM ********************
3440 REM *  JAHRESKOSTEN   *
3450 REM ********************
3460 D$=SEG$(DA$(DS),3,3)
3470 IF D$=".01" THEN 3480 E
LSE 3520
```

```
3480 IF BET$(DS)="" THEN BET
=0 :: GOTO 3500
3490 BET=VAL(BET$(DS))
3500 IF OP$(DS)="+" THEN MON
(1)=MON(1)+BET ELSE MON(1)=M
ON(1)-BET
3510 GOTO 4060
3520 IF D$=".02" THEN 3530 E
LSE 3570
3530 IF BET$(DS)="" THEN BET
=0 :: GOTO 3550
3540 BET=VAL(BET$(DS))
3550 IF OP$(DS)="+" THEN MON
(2)=MON(2)+BET ELSE MON(2)=M
ON(2)-BET
3560 GOTO 4060
3570 IF D$=".03" THEN 3580 E
LSE 3620
3580 IF BET$(DS)="" THEN BET
=0 :: GOTO 3600
3590 BET=VAL(BET$(DS))
3600 IF OP$(DS)="+" THEN MON
(3)=MON(3)+BET ELSE MON(3)=M
ON(3)-BET
3610 GOTO 4060
3620 IF D$=".04" THEN 3630 E
LSE 3670
3630 IF BET$(DS)="" THEN BET
=0 :: GOTO 3650
3640 BET=VAL(BET$(DS))
3650 IF OP$(DS)="+" THEN MON
(4)=MON(4)+BET ELSE MON(4)=M
ON(4)-BET
3660 GOTO 4060
3670 IF D$=".05" THEN 3680 E
LSE 3720
3680 IF BET$(DS)="" THEN BET
=0 :: GOTO 3700
3690 BET=VAL(BET$(DS))
3700 IF OP$(DS)="+" THEN MON
(5)=MON(5)+BET ELSE MON(5)=M
ON(5)-BET
3710 GOTO 4060
3720 IF D$=".06" THEN 3730 E
LSE 3770
3730 IF BET$(DS)="" THEN BET
=0 :: GOTO 3750
3740 BET=VAL(BET$(DS))
3750 IF OP$(DS)="+" THEN MON
(6)=MON(6)+BET ELSE MON(6)=M
ON(6)-BET
3760 GOTO 4060
3770 IF D$=".07" THEN 3780 E
LSE 3820
3780 IF BET$(DS)="" THEN BET
=0 :: GOTO 3800
3790 BET=VAL(BET$(DS))
3800 IF OP$(DS)="+" THEN MON
(7)=MON(7)+BET ELSE MON(7)=M
ON(7)-BET
3810 GOTO 4060
3820 IF D$=".08" THEN 3830 E
LSE 3870
3830 IF BET$(DS)="" THEN BET
=0 :: GOTO 3850
3840 BET=VAL(BET$(DS))
3850 IF OP$(DS)="+" THEN MON
(8)=MON(8)+BET ELSE MON(8)=M
ON(8)-BET
3860 GOTO 4060
3870 IF D$=".09" THEN 3880 E
LSE 3920
3880 IF BET$(DS)="" THEN BET
=0 :: GOTO 3900
3890 BET=VAL(BET$(DS))
3900 IF OP$(DS)="+" THEN MON
(9)=MON(9)+BET ELSE MON(9)=M
ON(9)-BET
3910 GOTO 4060
3920 IF D$=".10" THEN 3930 E
LSE 3970
3930 IF BET$(DS)="" THEN BET
=0 :: GOTO 3950
3940 BET=VAL(BET$(DS))
3950 IF OP$(DS)="+" THEN MON
(10)=MON(10)+BET ELSE MON(10
)=MON(10)-BET
3960 GOTO 4060
3970 IF D$=".11" THEN 3980 E
LSE 4020
3980 IF BET$(DS)="" THEN BET
=0 :: GOTO 4000
3990 BET=VAL(BET$(DS))
4000 IF OP$(DS)="+" THEN MON
(11)=MON(11)+BET ELSE MON(11
)=MON(11)-BET
4010 GOTO 4060
4020 IF D$=".12" THEN 4030 E
LSE 4060
4030 IF BET$(DS)="" THEN BET
=0 :: GOTO 4050
4040 BET=VAL(BET$(DS))
4050 IF OP$(DS)="+" THEN MON
(12)=MON(12)+BET ELSE MON(12
)=MON(12)-BET
4060 RETURN
4070 REM ********************
4080 REM *  JAHRESGRAPHIK  *
4090 REM ********************
4100 FOR I=1 TO 12 :: MO(I)=
MON(I):: NEXT I
4110 FOR I=1 TO 12 :: MO(I)=
INT(MO(I)):: NEXT I
4120 FOR I=1 TO 12
4130 IF MO(I)<=50 THEN MO(I)
=0
4140 IF MO(I)>50 AND MO(I)<=
150 THEN MO(I)=100/100 ELSE
IF MO(I)>150 AND MO(I)<=250
THEN MO(I)=200/100 ELSE IF M
O(I)>250 AND MO(I)<=350 THEN
 MO(I)=300/100 :: GOTO 4210
4150 IF MO(I)>350 AND MO(I)<
=450 THEN MO(I)=400/100 ELSE
 IF MO(I)>450 AND MO(I)<=550
 THEN MO(I)=500/100 ELSE IF
```

```
MO(I)>550 AND MO(I)<=650 THE
N MO(I)=600/100 :: GOTO 4210
4160 IF MO(I)>650 AND MO(I)<
=750 THEN MO(I)=700/100 ELSE
 IF MO(I)>750 AND MO(I)<=850
 THEN MO(I)=800/100 ELSE IF
MO(I)>850 AND MO(I)<=950 THE
N MO(I)=900/100 :: GOTO 4210
4170 IF MO(I)>950 AND MO(I)<
=1050 THEN MO(I)=1000/100 EL
SE IF MO(I)>1050 AND MO(I)<=
1150 THEN MO(I)=1100/100 ELS
E IF MO(I)>1150 AND MO(I)<=1
250 THEN MO(I)=1200/100 :: G
OTO 4210
4180 IF MO(I)>1250 AND MO(I)
<=1350 THEN MO(I)=1300/100 E
LSE IF MO(I)>1350 AND MO(I)<
=1450 THEN MO(I)=1400/100 EL
SE IF MO(I)>1450 AND MO(I)<=
1550 THEN MO(I)=1500/100 ::
GOTO 4210
4190 IF MO(I)>1550 AND MO(I)
<=1650 THEN MO(I)=1600/100 E
LSE IF MO(I)>1650 AND MO(I)<
=1750 THEN MO(I)=1700/100 EL
SE IF MO(I)>1750 AND MO(I)<=
1850 THEN MO(I)=1800/100 ::
GOTO 4210
4200 IF MO(I)>1850 AND MO(I)
<=1950 THEN MO(I)=1900/100 E
LSE IF MO(I)>1950 THEN MO(I)
=2000/100
4210 NEXT I
4220 CALL CLEAR :: CALL SCRE
EN(2)
4230 A1$="FFFFFFFFFFFFFFFF"
:: B1$="0103070F1F3F7FFF"
4240 CALL CHAR(104,A1$,112,A
1$,120,B1$,128,B1$,136,B1$):
: CALL COLOR(10,6,2,11,12,2,
12,2,12,13,16,2,14,12,16)
4250 FOR I=20 TO 1 STEP -1 :
: DISPLAY AT(23-I,1):I :: NE
XT I
4260 DISPLAY AT(1,4):"KOSTEN
 IN 100 DM" :: CALL COLOR(3,
16,2,4,16,2,5,16,2,6,16,2,7,
16,2,8,16,2)
4270 DISPLAY AT(24,4):"JAFEM
AAPMAJUJLAUSEOKNODE"
4280 IF SS=157 THEN 4290 ELS
E 4300
4290 M=0
4300 FOR P=6 TO 29 STEP 2
4310 M=M+1
4320 IF MO(M)=0 THEN 4350
4330 CALL VCHAR(23-MO(M),P,1
04,MO(M)):: CALL VCHAR(23-MO
(M),P+1,112,MO(M)-1):: CALL
HCHAR(22,P+1,120)
4340 CALL HCHAR(23-MO(M),P+1
,136):: CALL HCHAR(23-MO(M),
P,128)
4350 NEXT P
4360 CALL KEY(0,K,S):: IF S=
0 THEN 4360 ELSE 4370
4370 CALL CLEAR :: CALL SCRE
EN(3):: CALL COLOR(3,2,1,4,2
,1,5,2,1,6,2,1,7,2,1,8,2,1):
: GOTO 2600
4380 REM ********************
4390 REM * AUSGABE JAHRESK.*
4400 REM ********************
4410 DISPLAY AT(1,3)ERASE AL
L:USING "JANUAR....=>  #####.
## DM":MON(1)
4420 DISPLAY AT(3,3):USING "
FEBRUAR...=>  #####.## DM ":M
ON(2)
4430 DISPLAY AT(5,3):USING "
MAERZ.....=>  #####.## DM":MO
N(3)
4440 DISPLAY AT(7,3):USING "
APRIL.....=>  #####.## DM ":M
ON(4)
4450 DISPLAY AT(9,3):USING "
MAI.......=>  #####.## DM":MO
N(5)
4460 DISPLAY AT(11,3):USING
"JUNI......=>  #####.## DM":M
ON(6)
4470 DISPLAY AT(13,3):USING
"JULI......=>  #####.## DM":M
ON(7)
4480 DISPLAY AT(15,3):USING
"AUGUST....=>  #####.## DM":M
ON(8)
4490 DISPLAY AT(17,3):USING
"SEPTEMBER =>  #####.## DM":M
ON(9)
4500 DISPLAY AT(19,3):USING
"OKTOBER...=>  #####.## DM":M
ON(10)
4510 DISPLAY AT(21,3):USING
"NOVEMBER..=>  #####.## DM":M
ON(11)
4520 DISPLAY AT(23,3):USING
"DEZEMBER..=>  #####.## DM":M
ON(12)
4530 CALL KEY(0,K,S):: IF S=
0 THEN 4530 ELSE 2600
```

LISTINGS

UNSER TELEFONSERVICE

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite

ACHTUNG !!!

Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13

RECHTS-KOSTEN

Mit diesem Programm werden Rechtsanwaltskosten berechnet, die üblicherweise im Rahmen eines Zivilprozesses entstehen.
Anwender können sein: Rechtsanwälte bzw. Bürovorsteher sowie mit Kostenfestsetzung betraute Rechtspfleger bei Gerichten.
Neben Zeit- und Rechenersparnis bietet das Programm folgende Vorteile:

- im Dialog werden zunächst die einzelnen Gebührentatbestände bejaht oder verneint. Die Gefahr des "Vergessens" einer Gebühr ist damit praktisch gebannt;
- mit den üblichen Gebührentabellen lassen sich nur Gebühren aus einem Streitwert bis maximal 5 Mio. DM ablesen, für höhere Werte ist eine mehr oder weniger aufwendige Berechnung erforderlich. Das Computerprogramm berücksichtigt dagegen auch Werte über 5 Mio. DM;
- das Programm enthält sowohl die Tabelle für Regelgebühren als auch die Prozeßkostenhilfe-Tabelle, die ab einem Streitwert von über 5600,– DM niedrigere Gebührensätze festlegt;
- in der Praxis wird häufig übersehen, daß bei der Addition von Prozeß- und Differenzgebühr insgesamt nur höchstens eine volle Gebühr aus dem höheren Streitwert angesetzt werden darf. Das Programm berücksichtigt dies dagegen automatisch;
- Auf- bzw. Abrunden der Endsumme erledigt der Computer ebenso automatisch, wie er die Vorschrift des §26 Satz 3 BRAGO berücksichtigt, nach der bei den Postauslagen Pfennigbeträge stets auf volle 10 Pfennig aufzurunden sind.

Mit wenigen einfachen Zusatzstatements kann das Programm ergänzt werden, so daß über einen Drukker die fertigen Kostenrechnung oder der fertige Kostenfestsetzungsbeschluß erstellt wird.

Programmablauf:

a) Eingabe:
Zunächst wird festgelegt, ob die Gebühren nach der BRAGO- oder der PKH-Tabelle zu errechnen sind, und ob es sich um Kosten der 1., 2., oder 3. Instanz handelt.
Es folgt die Aufzählung der einzelnen Gebühren, die durch Eingaben der "1" bejaht oder durch "0" verneint werden, wobei – wenn die einzelne Gebühr bejaht wird – jeweils der Streitwert einzugeben ist.
Das gleiche geschieht mit den Auslagen: Schreibauslagen werden direkt eingegeben (pro Seite 1,– DM bzw. "0", wenn keine entstanden sind). Zur Fahrtkostenberechnung brauchen nur die gefahrenen Kilometer eingegeben werden, die Berechnung übernimmt der Computer. Werden keine Fahrtkosten geltend gemacht, ist wieder "0" einzugeben. Die Höhe des Abwesenheitsgeldes berechnet der Computer selbständig aufgrund der einzugebenden Stunden (oder "0", wenn nichts geltend gemacht wird).

b) Ausdruck auf dem Bildschirm:
Zunächst erscheint die Zeile: "Kosten berechnet nach" und danach entweder "BRAGO-Tabelle" oder "PKH-Tabelle" mit den einschlägigen Gesetzesvorschriften ("BRAGO Par." = BRAGO-Paragraphen).
Sodann werden die Gebührentatbestände mit den jeweils zugehörenden Streitwerten und Gebühren ausgedruckt.
Danach erscheint die Zeile "Taste ENTER drükken": hier stoppt der Ausdruck (wegen der nicht ausreichenden Zeilenkapazität des Bildschirmes), bis die Taste "ENTER" gedrückt wird.
Abschließend werden die Auslagen, soweit sie entstanden sind, Summe, Mehrwertsteuer und der Gesamtbetrag ausgedruckt.

LISTINGS

```
10 REM**********************
11 REM*RECHTSANWALTSKOST.*
12 REM*                    *
13 REM*  Copyright  by     *
14 REM*Dipl.-Rechtspfleg.*
15 REM*  Walter Platt      *
16 REM*                    *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                    *
23 REM* Speicherbelegung *
24 REM*    7789 Bytes     *
25 REM**********************
26 REM
150 CALL CLEAR
160 PRINT TAB(5);"berechnung
 von":TAB(5);"rechtsanwaltsk
osten":TAB(5);"(zivilprozess
)": : : :
170 PRINT "tabelle waehlen:"
:TAB(10);"brago=1":TAB(10);"
pkh  =2"
180 INPUT TABELLE
190 INPUT "instanz (1,2,3)?"
:INSTANZ
200 PRINT
210 PRINT "GEBUEHREN:":TAB(4
);"JA=1,NEIN=0"
220 PRINT
230 INPUT "prozessgeb.?":PG
240 IF PG<>1 THEN 320
250 GOSUB 920
260 IF INSTANZ<>1 THEN 300
270 PG=G(I)
280 SPG=S
290 GOTO 320
300 PG=G(I)/10*13
310 SPG=S
320 INPUT "volle verh./eroer
t.geb.?":VG
330 IF VG<>1 THEN 410
340 GOSUB 920
350 IF INSTANZ<>1 THEN 390
360 VG=G(I)
370 SVG=S
380 GOTO 410
390 VG=G(I)/10*13
400 SVG=S
410 INPUT "halbe verh.geb.?"
:HVG
420 IF HVG<>1 THEN 500
430 GOSUB 920
440 IF INSTANZ<>1 THEN 480
450 HVG=G(I)/2
460 SHVG=S
470 GOTO 500
480 HVG=G(I)/20*13
490 SHVG=S
500 INPUT "beweisgeb.?":BG
510 IF BG<>1 THEN 590
520 GOSUB 920
530 IF INSTANZ<>1 THEN 570
540 BG=G(I)
550 SBG=S
560 GOTO 590
570 BG=G(I)/10*13
580 SBG=S
590 INPUT "vergleichsgeb.?":
VGG
600 IF VGG<>1 THEN 680
610 GOSUB 920
620 IF INSTANZ<>1 THEN 660
630 VGG=G(I)
640 SVGG=S
650 GOTO 680
660 VGG=G(I)/10*13
670 SVGG=S
680 INPUT "differenzgeb.?":D
G
690 IF DG<>1 THEN 770
700 S=SVGG-SPG
710 SDG=S
720 GOSUB 940
730 IF INSTANZ<>1 THEN 760
740 DG=G(I)/2
750 GOTO 770
760 DG=G(I)/20*13
770 INPUT "schreibauslagen=
":SCHREIB
780 INPUT "fahrtkosten:km= "
:KM
790 FAHRT=KM*0.4
800 INPUT "abwesenh.in stund
en= ":STUN
810 IF STUN>0 THEN 840
820 ABW=0
830 GOTO 3510
840 IF STUN>4 THEN 870
850 ABW=20
860 GOTO 3510
870 IF STUN>8 THEN 900
880 ABW=40
890 GOTO 3510
900 ABW=75
910 GOTO 3510
920 INPUT "streitwert?":S
930 IF TABELLE=2 THEN 4040
940 IF S<>0 THEN 970
950 G(I)=0
960 GOTO 3500
970 IF S>200 THEN 1000
980 G(I)=30
990 GOTO 3500
1000 IF S>300 THEN 1030
1010 G(I)=40
1020 GOTO 3500
1030 IF S>500 THEN 1060
1040 G(I)=50
1050 GOTO 3500
1060 IF S>700 THEN 1090
1070 G(I)=60
1080 GOTO 3500
1090 IF S>900 THEN 1120
1100 G(I)=70
1110 GOTO 3500
1120 IF S>1200 THEN 1150
```

```
1130 G(I)=85
1140 GOTO 3500
1150 IF S>1600 THEN 1180
1160 G(I)=103
1170 GOTO 3500
1180 IF S>2000 THEN 1210
1190 G(I)=121
1200 GOTO 3500
1210 IF S>2400 THEN 1240
1220 G(I)=139
1230 GOTO 3500
1240 IF S>2800 THEN 1270
1250 G(I)=157
1260 GOTO 3500
1270 IF S>3200 THEN 1300
1280 G(I)=175
1290 GOTO 3500
1300 IF S>3600 THEN 1330
1310 G(I)=193
1320 GOTO 3500
1330 IF S>4000 THEN 1360
1340 G(I)=211
1350 GOTO 3500
1360 IF S>4400 THEN 1390
1370 G(I)=229
1380 GOTO 3500
1390 IF S>4800 THEN 1420
1400 G(I)=247
1410 GOTO 3500
1420 IF S>5200 THEN 1450
1430 G(I)=265
1440 GOTO 3500
1450 IF S>5600 THEN 1480
1460 G(I)=283
1470 GOTO 3500
1480 IF S>6400 THEN 1510
1490 G(I)=321
1500 GOTO 3500
1510 IF S>7200 THEN 1540
1520 G(I)=358
1530 GOTO 3500
1540 IF S>8000 THEN 1570
1550 G(I)=395
1560 GOTO 3500
1570 IF S>9000 THEN 1600
1580 G(I)=442
1590 GOTO 3500
1600 IF S>10000 THEN 1630
1610 G(I)=489
1620 GOTO 3500
1630 IF S>12000 THEN 1660
1640 G(I)=552
1650 GOTO 3500
1660 IF S>14000 THEN 1690
1670 G(I)=615
1680 GOTO 3500
1690 IF S>16000 THEN 1720
1700 G(I)=677
1710 GOTO 3500
1720 IF S>18000 THEN 1750
1730 G(I)=739
1740 GOTO 3500
1750 IF S>20000 THEN 1780
1760 G(I)=800
1770 GOTO 3500
1780 IF S>25000 THEN 1810
1790 G(I)=880
1800 GOTO 3500
1810 IF S>30000 THEN 1840
1820 G(I)=960
1830 GOTO 3500
1840 IF S>35000 THEN 1870
1850 G(I)=1040
1860 GOTO 3500
1870 IF S>40000 THEN 1900
1880 G(I)=1120
1890 GOTO 3500
1900 IF S>45000 THEN 1930
1910 G(I)=1200
1920 GOTO 3500
1930 IF S>50000 THEN 1960
1940 G(I)=1235
1950 GOTO 3500
1960 IF S>55000 THEN 1990
1970 G(I)=1270
1980 GOTO 3500
1990 IF S>60000 THEN 2020
2000 G(I)=1305
2010 GOTO 3500
2020 IF S>65000 THEN 2050
2030 G(I)=1340
2040 GOTO 3500
2050 IF S>70000 THEN 2080
2060 G(I)=1375
2070 GOTO 3500
2080 IF S>75000 THEN 2110
2090 G(I)=1410
2100 GOTO 3500
2110 IF S>80000 THEN 2140
2120 G(I)=1445
2130 GOTO 3500
2140 IF S>85000 THEN 2170
2150 G(I)=1480
2160 GOTO 3500
2170 IF S>90000 THEN 2200
2180 G(I)=1515
2190 GOTO 3500
2200 IF S>95000 THEN 2230
2210 G(I)=1550
2220 GOTO 3500
2230 IF S>100000 THEN 2260
2240 G(I)=1585
2250 GOTO 3500
2260 IF S>110000 THEN 2290
2270 G(I)=1605
2280 GOTO 3500
2290 IF S>120000 THEN 2320
2300 G(I)=1625
2310 GOTO 3500
2320 IF S>130000 THEN 2350
2330 G(I)=1640
2340 GOTO 3500
2350 IF S>140000 THEN 2380
2360 G(I)=1710
2370 GOTO 3500
2380 IF S>150000 THEN 2410
```

```
2390 G(I)=1780
2400 GOTO 3500
2410 IF S>160000 THEN 2440
2420 G(I)=1850
2430 GOTO 3500
2440 IF S>170000 THEN 2470
2450 G(I)=1920
2460 GOTO 2460
2470 IF S>180000 THEN 2500
2480 G(I)=1990
2490 GOTO 3500
2500 IF S>190000 THEN 2530
2510 G(I)=2060
2520 GOTO 3500
2530 IF S>200000 THEN 2560
2540 G(I)=2130
2550 GOTO 3500
2560 IF S>220000 THEN 2590
2570 G(I)=2250
2580 GOTO 3500
2590 IF S>240000 THEN 2620
2600 G(I)=2370
2610 GOTO 3500
2620 IF S>260000 THEN 2650
2630 G(I)=2490
2640 GOTO 3500
2650 IF S>280000 THEN 2680
2660 G(I)=2610
2670 GOTO 3500
2680 IF S>300000 THEN 2710
2690 G(I)=2730
2700 GOTO 3500
2710 IF S>320000 THEN 2740
2720 G(I)=2850
2730 GOTO 3500
2740 IF S>340000 THEN 2770
2750 G(I)=2970
2760 GOTO 3500
2770 IF S>360000 THEN 2800
2780 G(I)=3090
2790 GOTO 3500
2800 IF S>380000 THEN 2830
2810 G(I)=3210
2820 GOTO 3500
2830 IF S>400000 THEN 2860
2840 G(I)=3330
2850 GOTO 3500
2860 IF S>430000 THEN 2890
2870 G(I)=3450
2880 GOTO 3500
2890 IF S>460000 THEN 2920
2900 G(I)=3570
2910 GOTO 3500
2920 IF S>490000 THEN 2950
2930 G(I)=3690
2940 GOTO 3500
2950 IF S>520000 THEN 2980
2960 G(I)=3810
2970 GOTO 3500
2980 IF S>550000 THEN 3010
2990 G(I)=3930
3000 GOTO 3500
3010 IF S>580000 THEN 3040
```

LISTINGS

```
3020 G(I)=4050
3030 GOTO 3500
3040 IF S>610000 THEN 3070
3050 G(I)=4170
3060 GOTO 3500
3070 IF S>640000 THEN 3100
3080 G(I)=4290
3090 GOTO 3500
3100 IF S>670000 THEN 3130
3110 G(I)=4410
3120 GOTO 3500
3130 IF S>700000 THEN 3160
3140 G(I)=4530
3150 GOTO 3500
3160 IF S>730000 THEN 3190
3170 G(I)=4650
3180 GOTO 3500
3190 IF S>760000 THEN 3220
3200 G(I)=4770
3210 GOTO 3500
3220 IF S>790000 THEN 3250
3230 G(I)=4890
3240 GOTO 3500
3250 IF S>820000 THEN 3280
3260 G(I)=5010
3270 GOTO 3500
3280 IF S>850000 THEN 3310
3290 G(I)=5130
3300 GOTO 3500
3310 IF S>880000 THEN 3340
3320 G(I)=5250
3330 GOTO 3500
3340 IF S>910000 THEN 3370
3350 G(I)=5370
3360 GOTO 3500
3370 IF S>940000 THEN 3400
3380 G(I)=5490
3390 GOTO 3500
3400 IF S>970000 THEN 3430
3410 G(I)=5610
3420 GOTO 3500
3430 IF S>1000000 THEN 3460
3440 G(I)=5730
3450 GOTO 3500
3460 QUOTE=(S-1000000)/50000
3470 FAKTOR=INT(-QUOTE)
3480 FAKTOR=ABS(FAKTOR)
3490 G(I)=FAKTOR*150+5730
3500 RETURN
3510 CALL CLEAR
3520 PRINT "kosten berechnet
 nach:"
3530 IF TABELLE=1 THEN 3560
3540 PRINT "pkh-tabelle":"br
ago-par.121,123,126 sowie"
3550 GOTO 3570
3560 PRINT "brago-tabelle":"
brago-par.:"
3570 PRINT "11,23,25-28,31-3
3."
3580 PRINT
3590 IF PG<1 THEN 3610
3600 PRINT "proz.g:":"wert="
```

```
;SPG;"dm":TAB(12);PG;"DM"
3610 IF VG<1 THEN 3630
3620 PRINT "eroert.g bzw.":"
volle verh.g:":"wert=";SVG;"
dm":TAB(12);VG;"dm"
3630 IF HVG<1 THEN 3650
3640 PRINT "halbe verh.g:":"
wert:";SHVG;"dm":TAB(12);HVG
;"dm"
3650 IF BG<1 THEN 3670
3660 PRINT "bew.g:":"wert:";
SBG;"dm":TAB(12);BG;"dm"
3670 IF VGG<1 THEN 3690
3680 PRINT "vergl.g:":"wert:
";SVGG;"dm":TAB(12);VGG;"dm"
3690 IF DG<1 THEN 3740
3700 KON=VGG-PG
3710 IF DG<KON THEN 3730
3720 DG=KON
3730 PRINT "diff.g:":"wert:"
;SDG;"dm":TAB(12);DG;"dm":
3740 PRINT
3750 INPUT "taste ENTER drue
cken!":ETR$
3760 IF SCHREIB<1 THEN 3780
3770 PRINT :"schreibausl.=":
TAB(12);SCHREIB;"dm"
3780 IF FAHRT<0.4 THEN 3800
3790 PRINT "fahrtkosten:":KM
;"km*0.40 dm":TAB(12);FAHRT;
"dm"
3800 IF STUN<=0 THEN 3820
3810 PRINT "tagegeld:":STUN;
"stunden=":TAB(12);ABW;"dm"
3820 GESAMT=PG+VG+HVG+BG+VGG
+DG
3830 POST=GESAMT/100*15
3840 IF POST>40 THEN 3850 EL
SE 3860
3850 POST=40
3860 IF POST=0 THEN 3900
3870 IF 10*POST-INT(10*POST)
=0 THEN 3890
3880 POST=(INT(10*POST)+1)/1
0
3890 PRINT "postgeb.=":TAB(1
2);POST;"dm":
3900 PRINT
3910 SUMME=GESAMT+POST+SCHRE
IB+FAHRT+ABW
3920 PRINT "summe=";TAB(12);
SUMME;"dm"
3930 MWST=SUMME/100*14
3940 PRINT "14%mwst=";TAB(12
);MWST;"dm":"_______________
_____________"
3950 XX=SUMME+MWST
3960 IF 100*XX-INT(100*XX)=0
 THEN 4020
3970 XY=(100*XX)-INT(XX*100)
3980 IF XY<0.5 THEN 4010
3990 XX=(INT(100*XX)+1)/100
4000 GOTO 4020
```

LISTINGS

```
4010 XX=INT(XX*100)/100
4020 PRINT "gesamt:";TAB(12)
;XX;"dm":
4030 END
4040 IF S<=5600 THEN 940
4050 IF S>6400 THEN 4080
4060 G(I)=298
4070 GOTO 3500
4080 IF S>7200 THEN 4110
4090 G(I)=324
4100 GOTO 3500
4110 IF S>8000 THEN 4140
4120 G(I)=346
4130 GOTO 3500
4140 IF S>9000 THEN 4170
4150 G(I)=368
4160 GOTO 3500
4170 IF S>10000 THEN 4200
4180 G(I)=390
4190 GOTO 3500
4200 IF S>12000 THEN 4230
4210 G(I)=408
4220 GOTO 3500
4230 IF S>14000 THEN 4260
4240 G(I)=426
4250 GOTO 3500
4260 IF S>16000 THEN 4290
4270 G(I)=444
4280 GOTO 3500
4290 IF S>18000 THEN 4320
4300 G(I)=462
4310 GOTO 3500
4320 IF S>20000 THEN 4350
4330 G(I)=480
4340 GOTO 3500
4350 IF S>25000 THEN 4380
4360 G(I)=491
4370 GOTO 3500
4380 IF S>30000 THEN 4410
4390 G(I)=502
4400 GOTO 3500
4410 IF S>35000 THEN 4440
4420 G(I)=513
4430 GOTO 3500
4440 IF S>40000 THEN 4470
4450 G(I)=524
4460 GOTO 3500
4470 IF S>45000 THEN 4500
4480 G(I)=532
4490 GOTO 3500
4500 G(I)=540
4510 GOTO 3500
```

PRIVATE KLEINANZEIGEN KOSTENLOS!

SPEECH PROZESSOR 2

Dieses Programm erlaubt die Möglichkeiten des Speech Synthesizers, wie sie unter Extended-Basic gegeben sind, voll auszukosten. Sie werden sehen, daß der Speech Synthesizer mehr kann, als nur englisch zu sprechen.
Hier einige Beispiele, die Sie, nachdem Sie die Anleitung gelesen haben, einmal ausprobieren sollten. Die Beispiele müssen Sie dann unter B eingeben.

a) Kratzen einer Plattenspielernadel auf einer Schallplatte:
96,0,12,130,0,0,37,167,42,221,98,231,59,191,249

b) Das Klopfen an Glas:
96,0,7,55,0,0,37,167,42,221

c) Beispiel von konreten Namen:
96,0,26,0,0,0,0,0,43,215,180,167,51,219,217,206,119,190,243,81,212,34,132,93,75,102,213,171,90

Alle Beispiele müssen unter B (String erstellen) eingegeben werden.

Anmerkung:

Das Programm läuft sowohl mit Kassettenrekorder als auch mit Diskettenstation. Dies liegt daran, daß es mit seiner Länge von 11940 Bytes noch vom Kassettenloader geladen und gelesen werden kann.
Die Speichererweiterung wird aber unbedingt für die Variablen benötigt, die im Programm anfallen.
Erklärung, wie Sie die Speechstrings in ein Programm einbinden können, am Beispiel a:
Zuerst müssen Sie die Codes in Datazeilen abspeichern.

```
100 DATA 96,0,12,130,0,0,37,167,42,221,98,231,
    59,191,249
110 RESTORE 100 :: FOR I=1 TO 15 :: READ
    A$ :: B$=B$&CHR$(A$) :: NEXT I
```

Nach diesen beiden Zeilen steht Ihnen das Geräusch mit CALL SAY(,B$) zur Verfügung.

PROGRAMMBESCHREIBUNG:

Nach dem Programmstart steht Ihnen sofort das Hauptmenü zur Verfügung. Haben Sie keinen Speech Synthesizer oder keine RAM-Erweiterung angeschlossen, so erscheint eine Warnung des Programms und es bricht ab.

a) = STRING AUSSPRECHEN

Hier können Sie erstellte Speechstrings vom SPS. aussprechen lassen. Nach dem Drücken von A kommen Sie in ein Untermenü, in dem Sie wählen können, ob Sie mit A) einen erstellten String aussprechen lassen möchten oder ob Sie mit B) Wörter aus dem Wortschatz des SPS. eingeben möchten, die dann ausgesprochen werden.
Zu A) muß man noch anfügen, daß alle Speechstrings unter einem von Ihnen eingegebenen Namen abgespeichert werden. Diesen Namen müssen Sie eingeben, wenn Sie einen String bearbeiten oder sonstiges mit ihm machen möchten. Sie werden jeweils aufgefordert, den Namen einzugeben. Haben Sie einen unbekannten Namen eingegeben, so können Sie sich das Namensverzeichnis auflisten lassen. Die Tastenbelegungen sind jeweils angegeben.

b) = STRING EINGEBEN

Hier können Sie eigene Speechstrings eingeben. Nach der Anwahl von B gelangen Sie in ein Unterprogramm. Mit A) brauchen Sie erst beim vierten Code anfangen einzugeben, da die ersten drei Codes in der Regel gleich sind, nämlich CHRS(96)&CHRS(0)&CHRS (Anzahl der noch folgenden Codes). Mit B) können Sie auch die ersten drei Codes mit eingeben. Nach diesem Untermenü kehren Sie zur eigentlichen Eingabe zurück. Steht vor dem Rechteck in der Bildschirmmitte ein Pfeil, so können Sie Werte zwischen 0 und 255 eingeben. Steht kein Pfeil davor, so gelten Tastenbelegungen, die im unteren Teil des Bildschirms angegeben sind. Haben Sie hier FCTN 3 gedrückt, so werden Sie gefragt, ob Sie eine neue Eingabe machen möchten. Tippen Sie hier J ein, so können Sie einen neuen String eingeben, aber der alte geht verloren. Wenn Sie allerdings N eingeben, so können Sie den soeben eingegebenen String mit einem Namen belegen und

abspeichern. Es können maximal 20 Strings abgespeichert werden.

c) = STRING AUSGEBEN

Hier können Sie sich die ASCII-Werte eines abgespeicherten Strings ansehen. Auch hier müssen Sie zuerst den Namen des Strings eingeben. Danach wird oben auf dem Bildschirm der Name ausgegeben, darunter der String als Wort, darunter der erste ASCII-Code und daneben die Stelle, an der sich der Code gerade befindet. Am Anfang ist dieser Wert 1, da es sich um den ersten Code handelt. Mit den Tasten E und X können Sie nun die Codes „druchscrollen".

d) = STRINGS ANEINANDERHÄNGEN

Hier können Sie zwei Strings zu einem String verbinden. Dazu werden Sie aufgefordert, zuerst den Namen des Strings einzugeben, der später vorne stehen soll und danach den Namen des Strings einzugeben, der später hinten steht. Haben Sie dies getan, so erscheint der Kommentar, daß die Strings verbunden sind und Sie werden aufgefordert, einen Namen für den neuen String einzugeben. Anzumerken ist noch, daß der dritte Code des neuen Strings nicht angepaßt wurde (der Langencode).

e) = STRINGS VERÄNDERN

Hier können Sie nun Teile eines Strings verändern. Zuerst müssen Sie wieder den Namen eingeben. Dann werden Sie gefragt, ob Sie ein oder mehrere Zeichen eingeben möchten. Haben Sie auch dies getan, so werden Sie aufgefordert, einzugeben, an welcher Stelle Sie mit der Änderung beginnen möchten. Dazu sollten Sie sich unter STRING AUSGEBEN die Nummer aufschreiben, an der Sie z.B. einen Fehler entdeckt haben.

f) = TEILE DES STRINGS LÖSCHEN

Hier können Sie Teile eines Strings herauskürzen. Zuerst müssen Sie den Stringnamen eingeben und dann Anfangs- und Endnummer von dem Stück, das gelöscht werden soll, eingeben. Nehmen Sie für die Nummern c) STRING AUSGEBEN zu Hilfe. Haben Sie die Nummern eingegeben, können Sie wählen, ob die ersten drei Codes des neuen Strings angepaßt werden oder ob sie unverändert bleiben sollen. Danach können Sie den Namen für den neuen String eingeben.

g) = NAMENSVERZEICHNIS

Unter G wird das Namensverzeichnis der Strings ausgegeben, welches Sie auch anwählen können, wenn Sie einmal einen falschen Stringnamen eingegeben haben.

h) = WORT IN STRING UMWANDELN

Hier können Sie nun ein Wort aus dem Sprachschatz des SPS. eingeben und es in einen Speechstring mittels CAL SPGET umwandeln lassen. Danach müssen Sie den String wieder unter einem Namen abspeichern.

i) = ABSPEICHERN

Mit I können Sie die gesamten Speechstrings auf Kassette oder Diskette abspeichern. Zu beachten ist hier nur, daß aus systembedingten Gründen Speechstrings mit einer Länge größer als 191 Zeichen auf 191 Zeichen gekürzt werden müssen. Bei einem Diskettenlaufwerk werden sie vollständig abgespeichert.

j) = EINLESEN

Mit J können Sie die Speechstrings und deren Namen wieder einlesen.

k) = DRUCKEN

Mit K können Sie die Codes der Speechstrings und deren Namen ausdrucken.

l) = STRING LÖSCHEN

Hier können Sie einen String und dessen Namen löschen. Sie müssen lediglich den Namen des Strings eingeben, den Sie löschen möchten. Haben Sie einen String gelöscht, verschwindet dessen Name auch aus der Namensliste, und die nachfolgenden Namen rücken nach oben.

m)= STRING EINFÜGEN

Hier können Sie nun einen String in einen anderen String einschieben. Dazu geben Sie bitte zuerst den Namen des Strings an, in den ein String eingefügt werden soll und danach den Namen des Strings, den Sie einfügen möchten. Haben Sie dies getan, müssen Sie noch die Stelle angeben, an der der String eingefügt werden soll.
Ist dies alles geschehen, geben Sie bitte den Namen für den String ein.

n) = LÄNGE DES STRINGS

Mit N können Sie sich die Länge eines Strings ausgeben lassen.

o) = ENDE

O beendet das Programm.

Haben Sie schon 20 Strings erstellt und Sie wählen ein Unterprogramm an, in dem Sie einen neuen String erstellen, so erscheint eine Warnung und Sie müssen erst einen String löschen oder die Strings abspeichern und neu starten, bevor Sie einen neuen String erstellen können. Alle weiteren Funktionen und Tastenbelegungen, die hier nicht erklärt wurden, sind aus dem Programm ersichtlich.
Weiter empfehle ich, den ON ERROR-Befehl in Zeile 230 zu streichen, wenn Sie das Programm eintippen. Der On Error-Befehl überwacht die Druck-, die Abspeicher-, die Lade- und alle anderen Programmroutinen.
Bei der Druck-, der Abspeicher- und der Laderoutine erscheinen bei einem Fehler spezielle Fehlermeldungen und nach einem Tastendruck kehren Sie ins Hauptmenü zurück. Bei einem anderen ERROR kehren Sie sofort ins Hauptmenü zurück.
Das Programm ist, so weit es geht, absturzsicher. Dies wurde durch die ON ERROR-, die ON WARNING NEXT-, und die ON BREAK NEXT-Optionen erreicht. Weiter wurde auch die Quitttaste abgeschaltet.

Dirk Junghans

LISTINGS

```
10 ! **********************
11 ! *                    *
12 ! * SPEECHPROZESSOR 2  *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *   Dirk  Junghans   *
16 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! * Speechsynthesizer  *
23 ! * 32K-RAM-Erweiterung*
24 ! *  Kassettenrekorder *
25 ! *        oder        *
26 ! *  Diskettenstation  *
27 ! *  optional :Drucker *
28 ! *                    *
29 ! *  Speicherbelegung  *
30 ! *      11330 Bytes   *
31 ! *                    *
32 ! **********************
33 !
150 CALL PEEK(-31868,A,B)::
IF A=0 AND B=0 THEN 160 ELSE
 180
160 CALL CLEAR :: CALL SOUND
(200,220,0):: PRINT "WARNUNG
 !.....UEBERPR
UEFE SYSTEM !"
170 PRINT :: PRINT "KEINE RA
MERWEITERUNG ERKANNT !" :: S
TOP
180 CALL PEEK(-28276,A):: IF
 A<>0 THEN 210
190 CALL CLEAR :: CALL SOUND
(200,220,0):: PRINT "WARNUNG
 !.....UEBERPR
UEFE SYSTEM !"
200 PRINT :: PRINT "KEIN SPE
ECH SYNTHESIZER.....ERKANNT
!" :: STOP
210 CALL INIT :: CALL LOAD(-
31806,16)
220 CALL SCREEN(3):: FOR I=0
 TO 9 :: CALL COLOR(I,2,3)::
 NEXT I :: OPTION BASE 1
230 ON ERROR 1790 :: ON BREA
K NEXT :: ON WARNING NEXT
240 DIM SP$(20),NA$(20)
250 DATA A=STRING AUSSPRECHE
N,B=STRING EINGEBEN,C=STRING
 AUSGEBEN,D=STRINGS ANEINAND
ERHAENGEN,E=STRING VERAENDER
N
260 DATA F=TEILE DES STRINGS
 LOESCHEN,G=NAMENSVERZEICHNI
S,H=WORT IN STRING UMAENDERN
,I=ABSPEICHERN,J=EINLESEN,K=
DRUCKEN
270 DATA L=STRING LOESCHEN,M
=STRING EINFUEGEN,N=LAENGE D
ES STRINGS,O=ENDE
280 DISPLAY AT(1,1)ERASE ALL
:"SPEECHPROZESSOR 2.....
..(C)1986 BY DIRK JUNGHANS..
.............(DJIS)" :: DISP
LAY AT(4,1):RPT$("-",28)
290 RESTORE 250
300 FOR I=1 TO 15 :: READ A$
 :: DISPLAY AT(I+5,1):A$ ::
NEXT I :: DISPLAY AT(23,5):"
BITTE TASTE WAEHLEN"
310 CALL KEY(0,K,S):: IF K<6
5 OR K>79 OR S=0 THEN 310
320 IF K>70 THEN 340
330 ON K-64 GOTO 410,600,840
,950,1120,1380
340 ON K-70 GOTO 350,1500,36
0,370,380,1590,1640,1760,390
350 CALL INHALT(NA$()):: GOT
O 280
360 GOSUB 2090 :: GOTO 280
370 GOSUB 2180 :: GOTO 280
380 GOSUB 2270 :: GOTO 280
390 DISPLAY AT(10,1)ERASE AL
L:"BEENDEN ?(J/N)"
400 CALL KEY(0,K,S):: IF K=7
4 THEN PRINT "BYE !" :: STOP
 ELSE IF K=78 THEN 280 ELSE
400
410 DISPLAY AT(10,1)ERASE AL
L:"<A> GESPEICHERTEN STRING.
.......AUSSPRECHEN.........
...<B> WOERTER EINGEBEN UND.
.......AUSSPRECHEN"
420 CALL KEY(0,K,S):: IF S=0
 OR K<65 OR K>66 THEN 420
430 IF K=66 THEN 510
440 GOSUB 1980 :: IF NM=0 TH
EN 280
450 DISPLAY AT(20,1)ERASE AL
L:"<SPACE> AUSSPRECHEN......
...<FCTN 8> NEUE EINGABE....
...<FCTN 9> ZURUECK"
460 DISPLAY AT(8,1):"NAME DE
S STRINGS :" :: DISPLAY AT(1
0,1):NA$(NM)
470 CALL KEY(0,K,S):: IF K<>
32 THEN 490
480 CALL SAY(,SP$(NM)):: GOT
O 470
490 IF K<>15 THEN 500 ELSE G
OTO 280
500 IF K<>6 THEN 470 ELSE 41
0
510 DISPLAY AT(10,1):"BITTE
ZEICHENFOLGE EINGEBEN.ES SIN
D ALLE ZEICHEN.....ERLAUB
T, DIE DER SP.SY. VER-ARBEIT
EN KANN.(#.+-)"
520 DISPLAY AT(14,1):"AUSSER
 DEM KOMMA (,)"
530 DISPLAY AT(14,1):"EINGAB
E MAX 5 ZEILEN." :: INPUT S$
540 DISPLAY AT(10,1)ERASE AL
L:S$
```

```
550 DISPLAY AT(20,1):"<SPACE
> AUSSPRECHEN.....<FCTN
8> NEUE EINGABE.....<FCTN
9> ZURUECK"
560 CALL KEY(0,K,S):: IF K<>
32 THEN 580
570 CALL SAY(S$):: GOTO 560
580 IF K<>15 THEN 590 ELSE G
OTO 280
590 IF K<>6 THEN 560 ELSE 41
0
600 GOSUB 1940
610 W$="" :: DISPLAY AT(2,1)
ERASE ALL:"ES KOENNEN NUR WE
RTE VON....0 BIS 255 EINGEGE
BEN WERDEN."
620 DISPLAY AT(10,1):"<A> DI
E ERSTEN DREI KODES.....AU
TOMATISCH ERSTELLEN.....LA
SSEN.....<B> EI
NGABE DER ERSTEN DREI KODES"
630 DISPLAY AT(14,1):"....VO
N HAND"
640 CALL KEY(0,K,S):: IF S=0
 OR K<65 OR K>66 THEN 640
650 IF K=65 THEN VH=0 ELSE V
H=1
660 DISPLAY AT(10,1):RPT$("
",255)
670 DISPLAY AT(10,10):"-----
" :: DISPLAY AT(11,10):"|
|" :: DISPLAY AT(12,10):"---
---"
680 DISPLAY AT(20,10):"<ENTE
R>NAECHSTE EINGABE.....<FCT
N 3>EINGABE BEENDEN"
690 IF VH=0 THEN ZAE=4 ELSE
ZAE=1
700 DISPLAY AT(10,17):ZAE
710 DISPLAY AT(11,8)SIZE(2):
"->" :: ACCEPT AT(11,11)SIZE
(-3):W :: IF W<0 OR W>255 TH
EN 710 ELSE W$=W$&CHR$(W)
720 DISPLAY AT(11,8)SIZE(2):
"  "
730 FOR I=1 TO 50 :: NEXT I
740 CALL KEY(0,K,S):: IF S=0
 THEN 740
750 IF K<>13 THEN 770
760 ZAE=ZAE+1 :: GOTO 700
770 IF K<>7 THEN 740 ELSE DI
SPLAY AT(20,1):"NEUE EINGABE
 ?(J/N)"&RPT$(" ",50)
780 CALL KEY(0,K,S):: IF K=7
4 THEN 610 ELSE IF K=78 THEN
 790 ELSE 780
790 AN=AN+1
800 GOSUB 1920
810 IF VH=1 THEN 830
820 SP$(AN)=CHR$(96)&CHR$(0)
&CHR$(LEN(W$))&W$ :: GOTO 28
0
830 SP$(AN)=W$ :: GOTO 280
840 GOSUB 1980 :: IF NM=0 TH
EN 280
850 DISPLAY AT(3,1)ERASE ALL
:RPT$("-",28):: DISPLAY AT(1
4,1):RPT$("-",28)
860 DISPLAY AT(2,1):NA$(NM):
: DISPLAY AT(4,1):SP$(NM)
870 DISPLAY AT(21,1):"<X>VOR
WAERTS.....<E>RUE
CKWAERTS.....<FCTN
8>NOCHMAL.....<FCTN
9>ZURUECK"
880 FOR I=1 TO LEN(SP$(NM)):
: DISPLAY AT(16,10):ASC(SEG$
(SP$(NM),I,1)):: DISPLAY AT(
16,20):I
890 CALL KEY(0,K,S):: IF K=1
5 THEN 280 ELSE IF K=6 THEN
840
900 IF K=88 THEN 930
910 IF K<>69 THEN 890
920 I=I-2 :: IF I<0 THEN I=0
930 IF I=LEN(SP$(NM))THEN I=
LEN(SP$(NM))-1
940 NEXT I
950 GOSUB 1940 :: CALL CLEAR
960 DISPLAY AT(10,1):"BITTE
GEBEN SIE NACHDEM SIE EINE T
ASTE GEDRUECKT HABEN  DIE BE
IDEN NAMEN DER STRINGSEIN, D
IE SIE VERBINDEN.....MOECHT
EN."
970 DISPLAY AT(18,1):"BITTE
TASTE DRUECKEN "
980 CALL KEY(0,K,S):: IF S=0
 THEN 980
990 CALL MAGNIFY(2):: CALL C
LEAR :: CALL SPRITE(#1,49,2,
80,120,#2,46,2,80,128):: DIS
PLAY AT(12,18):"NAME"
1000 FOR I=1 TO 350 :: NEXT
I :: CALL DELSPRITE(ALL):: G
OSUB 1980 :: IF NM=0 THEN 28
0
1010 NM1=NM :: CALL CLEAR ::
 CALL SPRITE(#1,50,2,80,120,
#2,46,2,80,129):: DISPLAY AT
(12,18):"NAME" :: FOR I=1 TO
 350 :: NEXT I
1020 CALL DELSPRITE(ALL):: G
OSUB 1980 :: IF NM=0 THEN 28
0
1030 NM2=NM
1040 CALL CLEAR :: DISPLAY A
T(8,1):"NAME 1 :"¦NA$(NM1)¦"
 UND" :: DISPLAY AT(10,1):"N
AME 2 :"¦NA$(NM2):: DISPLAY
AT(12,1):"SIND VERBUNDEN."
1050 NM$=SP$(NM1)&SP$(NM2)
1060 AN=AN+1
1070 GOSUB 1920
1080 SP$(AN)=NM$
1090 DISPLAY AT(19,10):RPT$(
```

```
" ",140)
1100 DISPLAY AT(19,1):"<FCTN
 8> NOCHMAL.............<FCNT
 9> ZURUECK"
1110 CALL KEY(0,K,S):: IF K=
6 THEN 950 ELSE IF K=15 THEN
 280 ELSE 1110
1120 GOSUB 1940 :: GOSUB 198
0 :: IF NM=0 THEN 280
1130 DISPLAY AT(10,1)ERASE A
LL:"<A> EIN ZEICHEN AENDERN.
.............................
....<B> MEHRERE ZEICHEN AEND
ERN"
1140 CALL KEY(0,K,S):: IF K=
65 THEN 1150 ELSE IF K=66 TH
EN 1230 ELSE 1140
1150 CALL CLEAR :: DISPLAY A
T(5,1):"NAME :";NA$(NM)
1160 DISPLAY AT(10,1):"AN WI
EVIELTER STELLE STEHT  DAS Z
EICHEN, DAS SIE AEN-...DERN
MOECHTEN ? :"
1170 ACCEPT AT(12,18)SIZE(3)
VALIDATE(DIGIT):A :: IF A>25
5 THEN 1170
1180 DISPLAY AT(15,1):"DAS Z
EICHEN BESITZT DEN.....ASCII
-CODE :";ASC(SEG$(SP$(NM),A,
1)):: DISPLAY AT(18,1):"NEUE
R CODE :"
1190 ACCEPT AT(18,13)SIZE(3)
VALIDATE(DIGIT):B :: IF B>25
5 THEN 1190
1200 A$=SEG$(SP$(NM),1,A-1)&
CHR$(B)&SEG$(SP$(NM),A+1,LEN
(SP$(NM))-A)
1210 AN=AN+1 :: GOSUB 1920 :
: SP$(AN)=A$ :: DISPLAY AT(1
9,1):RPT$(" ",112):: DISPLAY
 AT(23,1):"<FCTN 8>NOCHMAL..
.........<FCTN 9>ZURUECK"
1220 CALL KEY(0,K,S):: IF K=
6 THEN 1120 ELSE IF K=15 THE
N 280 ELSE 1220
1230 DISPLAY AT(5,1)ERASE AL
L:"NAME :";NA$(NM)
1240 DISPLAY AT(8,1):"AN WIE
VIELTER STELLE.....MOECHT
EN SIE MIT DER AENDE- RUNG B
EGINNEN ? "
1250 SZ$="" :: ACCEPT AT(10,
18)SIZE(3)VALIDATE(DIGIT):A
:: B=A :: IF A>255 THEN 1250
1260 DISPLAY AT(8,1):RPT$("
",112):: DISPLAY AT(22,1):"<
ENTER>NAECHSTE EINGABE.....<
FCNT 8>NEUE STARTADRESSSE  <
FCTN 3>EINGABE BEENDEN"
1270 DISPLAY AT(12,1):"ALTER
 CODE :" :: DISPLAY AT(14,1)
:"EINGABE :" :: DISPLAY AT(1
2,13):ASC(SEG$(SP$(NM),A,1))
1280 DISPLAY AT(10,1):"NUMME
R :" :: DISPLAY AT(10,9):A
1290 ACCEPT AT(14,10)SIZE(3)
VALIDATE(DIGIT):WE :: IF WE>
255 THEN 1290
1300 SZ$=SZ$&CHR$(WE)
1310 CALL KEY(0,K,S):: IF K<
>13 THEN 1340
1320 A=A+1 :: IF A>LEN(SP$(N
M))THEN A=A-1
1330 GOTO 1270
1340 IF K=6 THEN 1230 ELSE I
F K=7 THEN 1350 ELSE 1310
1350 A$=SEG$(SP$(NM),1,B-1)&
SZ$&SEG$(SP$(NM),A+1,LEN(SP$
(NM))-A):: AN=AN+1 :: SP$(AN
)=A$ :: DISPLAY AT(22,1):RPT
$(" ",84):: GOSUB 1920
1360 DISPLAY AT(19,1):RPT$("
 ",168):: DISPLAY AT(23,1):"
<FCTN 8>NOCHMAL.............
<FCTN 9>ZURUECK"
1370 CALL KEY(0,K,S):: IF K=
6 THEN 1230 ELSE IF K=15 THE
N 280 ELSE 1370
1380 GOSUB 1940 :: GOSUB 198
0 :: IF NM=0 THEN 280
1390 DISPLAY AT(9,1)ERASE AL
L:"WELCHER TEIL DES STRINGS
   SOLL GELOESCHT WERDEN. BI
TTEANFANGS UND ENDNUMMER EIN
-  GEBEN."
1400 DISPLAY AT(14,1):NA$(NM
):: DISPLAY AT(16,1):"VON>..
.< BIS>...<"
1410 ACCEPT AT(16,5)SIZE(3):
N1 :: ACCEPT AT(16,14)SIZE(3
):N2
1420 IF (N1<1)OR(N2<1)OR(N1>
LEN(SP$(NM)))OR(N2>LEN(SP$(N
M)))OR(N1>N2)THEN 1400
1430 AN=AN+1 :: N0$=SEG$(SP$
(NM),1,N1-1)&SEG$(SP$(NM),N2
+1,LEN(SP$(NM))-N2):: SP$(AN
)=N0$
1440 DISPLAY AT(10,1)ERASE A
LL:"<1> DIE ERSTEN DREI KODE
S SO....LASSEN WIE SIE SIND.
 ...<2> DIE ERSTEN DREI KODE
S.......ANPASSEN."
1450 CALL KEY(0,K,S):: IF K=
49 THEN 1470 ELSE IF K=50 TH
EN 1460 ELSE 1450
1460 N0$=CHR$(96)&CHR$(0)&CH
R$(LEN(SP$(AN))-3)&SEG$(SP$(
AN),4,LEN(SP$(AN))-3):: SP$(
AN)=N0$
1470 GOSUB 1920 :: DISPLAY A
T(20,1):RPT$(" ",5*28)
1480 DISPLAY AT(23,1):"<FCTN
 8>NOCHMAL.............<FCTN
 9>ZURUECK"
1490 CALL KEY(0,K,S):: IF K=
```

```
6 THEN 1380 ELSE IF K=15 THE
N 280 ELSE 1490
1500 GOSUB 1940
1510 DISPLAY AT(7,1)ERASE AL
L:"BITTE DAS WORT EINGEBEN,
DASUMGEWANDELT WERDEN SOLL.
.....(NUR IM SP GESPEICHERTE
)"
1520 DISPLAY AT(10,1):"WOERT
ER WERDEN KORREKT......VERAR
BEITET."
1530 CALL HCHAR(14,2,62):: A
CCEPT AT(14,1):WE$ :: IF WE$
="" THEN 1530
1540 DISPLAY AT(16,1):"BITTE
 EINE SEKUNDE WARTEN !" :: A
N=AN+1 :: CALL SPGET(WE$,SP$
(AN))
1550 GOSUB 1920
1560 DISPLAY AT(19,10):RPT$(
" ",140)
1570 DISPLAY AT(19,1):"<FCTN
 8> NOCHMAL.....<FCNT
 9> ZURUECK"
1580 CALL KEY(0,K,S):: IF K=
6 THEN 1500 ELSE IF K=15 THE
N 280 ELSE 1580
1590 GOSUB 1980 :: IF NM=0 T
HEN 280
1600 DISPLAY AT(10,1)ERASE A
LL:NAME$;" IST GELOESCHT" ::
 NA$(NM)=""
1610 FOR I=NM+1 TO 15 :: NA$
(I-1)=NA$(I):: NEXT I :: DIS
PLAY AT(15,1):"BITTE TASTE D
RUECKEN FUER...RUECKKEHR !"
1620 FOR I=AN TO 15 :: NA$(I
)="" :: NEXT I :: AN=AN-1
1630 CALL KEY(0,K,S):: IF S=
0 THEN 1630 ELSE 280
1640 GOSUB 1940 :: CALL CLEA
R
1650 DISPLAY AT(10,1):"IN WE
LCHEN STRING SOLL EIN  STRIN
G EINGEFUEGT WERDEN ?" :: DI
SPLAY AT(15,1):"BITTE TASTE
DRUECKEN"
1660 CALL KEY(0,K,S):: IF S=
0 THEN 1660 ELSE GOSUB 1980
1670 IF NM=0 THEN 280
1680 NM1=NM :: DISPLAY AT(10
,1)ERASE ALL:"WELCHER STRING
 SOLL EINGE-  FUEGT WERDEN ?
" :: DISPLAY AT(15,1):"BITTE
 TASTE DRUECKEN"
1690 CALL KEY(0,K,S):: IF S=
0 THEN 1690 ELSE GOSUB 1980
1700 IF NM=0 THEN 280
1710 DISPLAY AT(10,1)ERASE A
LL:"AN WELCHER STELLE, SOLL
DER STRING EINGEFUEGT.......
....WERDEN ? :"
1720 ACCEPT AT(12,11)SIZE(3)
VALIDATE(DIGIT):STE :: IF ST
E=0 OR STE>LEN(SP$(NM1))THEN
 1720
1730 A$=SEG$(SP$(NM1),1,STE)
&SP$(NM)&SEG$(SP$(NM1),STE+1
,LEN(SP$(NM1))-STE):: AN=AN+
1 :: GOSUB 1920 :: SP$(AN)=A
$
1740 DISPLAY AT(19,1):RPT$("
",168):: DISPLAY AT(23,1):"
<FCTN 8>NOCHMAL.....
<FCTN 9>ZURUECK"
1750 CALL KEY(0,K,S):: IF K=
15 THEN 280 ELSE IF K=6 THEN
 1640 ELSE 1750
1760 GOSUB 1980 :: DISPLAY A
T(10,1)ERASE ALL:"DIE LAENGE
 FUER.....STRING : "
;NA$(NM):"BETRAEGT :";LEN(SP
$(NM));" ZEICHEN."
1770 DISPLAY AT(23,1):"<FCTN
 8>NOCHMAL.....<FCTN
 9>ZURUECK"
1780 CALL KEY(0,K,S):: IF K=
15 THEN 280 ELSE IF K=6 THEN
 1760 ELSE 1780
1790 CALL ERR(A1,A2,A3,A4)
1800 IF A1<>56 THEN 1820 ELS
E PRINT
1810 PRINT :"SPEECH-STRING Z
U LANG !" :: GOTO 1890
1820 IF (A1=130 OR A1=109)AN
D(A4>10030 AND A4<10090)THEN
 1830 ELSE 1840
1830 PRINT :: PRINT :"ERROR
BEIM ABSPEICHERN !" :: GOTO
1890
1840 IF (A1=130 OR A1=109)AN
D(A4>=10140 AND A4<=10180)TH
EN 1850 ELSE 1860
1850 PRINT :: PRINT :"ERROR
BEIM EINLESEN !" :: GOTO 189
0
1860 IF A1=130 AND(A4>=10240
 AND A4<=10270)THEN 1870 ELS
E 1880
1870 PRINT :: PRINT :"ERROR
BEIM AUSDRUCKEN !" :: GOTO 1
890
1880 ON ERROR 1790 :: RETURN
 NEXT
1890 PRINT "TASTE DRUECKEN"
1900 CALL KEY(0,K,S):: IF S=
0 THEN 1900
1910 ON ERROR 1790 :: RETURN
 280
1920 DISPLAY AT(19,10):"BITT
E NAME FUER DEN NEUEN...STRI
NG EINGEBEN." :: ACCEPT AT(2
2,1)SIZE(10):NA$(AN):: IF NA
$(AN)="" THEN 1920
1930 DISPLAY AT(23,1):"ES IS
T DER";AN;". STRING !" :: FO
```

```
R I=1 TO 300 :: NEXT I :: RE
TURN
1940 IF AN>=20 THEN 1950 ELS
E RETURN
1950 PRINT :: PRINT :"KEIN S
PEICHER FUER WEITERE  STRING
S VORHANDEN !" :: PRINT :"SI
E MUESSEN ERST EINEN......ST
RING LOESCHEN."
1960 PRINT :"BITTE TASTE DRU
ECKEN !"
1970 CALL KEY(0,K,S):: IF S=
0 THEN 1970 ELSE 280
1980 DISPLAY AT(10,1)ERASE A
LL:"BITTE NAME DES GEWUENSCH
TEN  STRINGS EINGEBEN:" :: A
CCEPT AT(11,19)SIZE(10):NAME
$ :: IF NAME$="" THEN 1980
1990 FOR I1=1 TO 20 :: IF NA
ME$<>NA$(I1)THEN 2010
2000 NM=I1 :: RETURN
2010 NEXT I1 :: DISPLAY AT(1
5,6):"NAME UNBEKANNT !"
2020 DISPLAY AT(20,1):"<FCTN
 7> NAMENSVERZEICHNIS  <FCTN
 8> NEUE EINGABE.......<FCTN
 9> ZURUECK"
2030 CALL KEY(0,K,S):: IF S=
0 THEN 2030
2040 IF K<>15 THEN 2060
2050 NM=0 :: RETURN
2060 IF K<>1 THEN 2080
2070 CALL INHALT(NA$()):: CA
LL CLEAR :: FOR I=1 TO 100 :
: NEXT I :: GOTO 2020
2080 IF K=6 THEN 1980 ELSE 2
030
2090 DISPLAY AT(10,1)ERASE A
LL:"ABSPEICHERN ?(J/N)"
2100 CALL KEY(0,K,S):: IF K=
78 THEN RETURN ELSE IF K<>74
 THEN 2100
2110 DISPLAY AT(12,1):"0=CS1
  1=DSK1 :" :: ACCEPT AT(12,
16)SIZE(1)VALIDATE("01"):A
2120 IF A=0 THEN 2130 ELSE D
ISPLAY AT(15,1):"NAME :" ::
ACCEPT AT(15,7)SIZE(10):AA$
2130 IF A=0 THEN OPEN #1:"CS
1",INTERNAL,OUTPUT,FIXED 192
 ELSE OPEN #1:"DSK1.AA$",INT
ERNAL,OUTPUT,FIXED 254
2140 PRINT #1:AN :: FOR I=1
TO AN
2150 IF A=0 THEN PRINT #1:SE
G$(SP$(I),1,191)ELSE PRINT #
1:SP$(I)
2160 PRINT #1:NA$(I)
2170 NEXT I :: CLOSE #1 :: R
ETURN
2180 DISPLAY AT(10,1)ERASE A
LL:"EINLESEN ?(J/N)"
2190 CALL KEY(0,K,S):: IF K=
78 THEN RETURN ELSE IF K<>74
 THEN 2190
2200 DISPLAY AT(12,1):"0=CS1
  1=DSK1 :" :: ACCEPT AT(12,
16)SIZE(1)VALIDATE("01"):A
2210 IF A=0 THEN 2220 ELSE D
ISPLAY AT(15,1):"NAME :" ::
ACCEPT AT(15,7)SIZE(10):AA$
2220 IF A=0 THEN OPEN #1:"CS
1",INTERNAL,INPUT ,FIXED 192
 ELSE OPEN #1:"DSK1.AA$",INT
ERNAL,INPUT ,FIXED 254
2230 INPUT #1:AN :: FOR I=1
TO AN
2240 INPUT #1:SP$(I)
2250 INPUT #1:NA$(I)
2260 NEXT I :: CLOSE #1 :: R
ETURN
2270 DISPLAY AT(10,1)ERASE A
LL:"DRUCKEN ?(J/N)"
2280 CALL KEY(0,K,S):: IF K=
78 THEN RETURN ELSE IF K<>74
 THEN 2280
2290 DISPLAY AT(10,1):"FILEN
AME :" :: ACCEPT AT(10,12):D
R$ :: IF DR$="" THEN 2290
2300 OPEN #2:DR$
2310 FOR I=1 TO AN
2320 PRINT #2:NA$(I)
2330 FOR I1=1 TO LEN(SP$(I))
:: A=ASC(SEG$(SP$(I),I1,1))
2340 PRINT #2:A;" ";
2350 NEXT I1 :: PRINT #2
2360 NEXT I :: CLOSE #2 :: R
ETURN
2370 SUB INHALT(NA$()):: CAL
L CLEAR
2380 DISPLAY AT(23,1):"<SPAC
E>WEITERBLAETTERN.......<FCT
N 9>ZURUECK"
2390 DISPLAY AT(1,1):"NAMENS
VERZEICHNIS" :: GOSUB 2430
2400 CALL KEY(0,K,S):: IF K=
15 THEN SUBEXIT ELSE IF K<>3
2 THEN 2400
2410 IF Z=1 THEN GOSUB 2440
ELSE IF Z=0 THEN GOSUB 2430
2420 GOTO 2400
2430 Z=1 :: FOR I=1 TO 10 ::
 DISPLAY AT(I*2+1,1):I;"=>";
NA$(I):: NEXT I :: RETURN
2440 Z=0 :: FOR I=11 TO 20 :
: DISPLAY AT(I*2-19,1):I;"=>
";NA$(I):: NEXT I :: RETURN
2450 SUBEND
```

BÖRSE

Basic für Anfänger. Orig. Cass. f. nur 6,50 DM inkl. Porto, Tel: 06638/1503

TI 99/4A 200,–, Musik-M. 40,–, Car-Wars 25,–, Alpiner 25,: Amaziing 30,–, Tomstonecity 25,–, Indoor-Soccer 30,–, Hopper 25,–, alle Module im guten Zustand. V.Kass.Progr. u. Ext. Basic. Tauschmöglk.!!! Computerkurs 67 Hefte 100,– DM. Ab 18 Uhr, Tel: 08841/40346

Verk. Ex-Basic + Progr. 100,– DM. Mini Mem. + Progr. 100,– DM. E/A 80,– DM. TE II 40,– DM. W. König, Neuffenstr. 16, 7317 Wendlingen

---

Verk. wegen Systemw. TI 99/4A + neuw. Ext. B. + 2 Module + Cass. Rec. Kabel + Joystick + Progr. Cass. + Lit. 500,– DM VB. Herwart Feuerstein, Konrad-Zirkel-Str. 18, 8788 Bad-Brückenau

---

Verk. TI 100,–, Ex-Basic 150,–, TI-Sprachsynt. 50,–, Tipbox 400,–, TI-RS 232 200'-, TI-Disk 400,–, 32 K 200,–, TI-Par. Drucker 50,–, TI-Joystick 30,–, Literatur 100,– DM. Auch kompl. 1200,– DM. H. Lehmann, Im Hasengrund 33c, 6101 Bickenbach

---

ENDLICH!!!
TI99/4A-gesteuertes Comp. Telefon! Idela für Mailboxbetrieb, Terminalbetrieb und Leute, die sich nicht gern die Finger wund wählen!!! Genaues erfahren sie unter 030/4013929 (ab 16.30 Uhr) od. mit frankiertem Rückumschl. an P. Siegel, Huttenstr. 27, 1000 Berlin 28

Echte Kleinbuchstaben für TI99/4A mit UCSD p-System. US und deutsche Zeichen!!! Konfort. Menüprogr. zur Schriftwahl. 5,25" Disk Code-file + 3 Datafiles: 20,– DM. Mit Quelltext 25,– DM. Näheres unt. Tel: 0511/889046

Verk. Centronic-TI-Schnittstelle - Preis VB evtl. Tausch geg. Module. Tel: 07156/5190. Meyer-Staufenbiel

---

Verk. TI 99/4A mit Handb. + Netzt. + Palmod. + Rec.Kabel + Progr. + Lit. VB 450,–, Ext. Basic II Plus + dt. Handbuch VB 260,– DM. Noch Garantie. Gerd Fleischmann, Norderneystr. 18, 4350 Recklinghausen

---

Buchungsjournal (mod.) 70'-, Datenverw. + Anal. (Mod.) 50,–, Lagerverwl. (Disk) 50,–, Rechnungsstellung (Disk) 50,–, Statistik (Mod.) 30,–, Vokabeltrai. (Cass.) 10,–, TI-Basic Kurs (engl. Cass.) 10,– DM. R. Unruh, Tel: 0711/719267

Verk. TI-Disk-Contr. 350,–. Disk-Manag. 2 20,–, Laufw. SSSD 200,–, 32 K RAM-Karte 200,– DM. Adr.: J. Ziegler, Rathausstr. 16, 7635 schwanau. Tel: 07324/3343

Verkaufe meinen TI 99/4A Konsole + Peri-Box m. Laufwerk + Disc-Manag. Modul + orig. X-Basic + Schnittst.-Karte f. P-Box (parallel/seriell) + Personal Rec. Keeping Modul + Statistic Modul f. nur 1100,– DM. Tel: 064 03/71104, o. 3551

Hallo!!!
Wer tauscht dt. Handb. für Ex-Basic geg engl. Handb. für Ex. Basic od. geg. Alien Addition Modul Angebote an: Ralf Naumann, Keplerstr. 21, 6680 Neunkirchen 7

TI-Adventure
5 neue Adv. f. TI-Adv. Modul. 40 % mehr Umfang, als orig. TI-Spiele. Jedes Spiel nur 15,– DM. öS 100, excl. Datenträger. Infos von: Thomas Catarozzi, Spaisingerstr. 42, A-1130 Wien

---

Verk. Exb. Handbuch (engl.) 20,– Sp. Ler. Arb. Buch 15,–, Monitorkabel 10,–, Buch mit 100 Spielen 10,–. Kass. m. 30 Spielen 10,–, Basiclehrgang auf Kass. 5,– DM. Tel: 0231/806525

Wer möchte eine 1,5 Seiten lange Poke- u. Peekliste haben? Schickt einfach 10,– DM-Schein und Rückporto an: Klaus J. Höltgen, Westerdorfstr. 10, 4300 Essen 12

Verk. Zusatzinfo f. E/A Buch mit/ohne Disk. Ausf. (Bitdarst.) u. einfache Bsp. für Assemblereinstieg o. Vorkenntnisse geeignet. Ab Mitte Aug. fert. Probeseiten bei Rückporto mögl. A. Pack, Eickelerstr. 60, 4690 Herne 2

Verk. Matrixdrucker M1009 (Brother) Schnittstellen: RS232 u. Centronics. Guter Zustand, wenig gebraucht. Preis: VB 450,–. Tel: 06181/51236, B. Kronenberger

---

Verk. 32K intern orig. TI 200,–
Module: Ext. Basic (TI) 100,–; Mini Mem 100,–, TI-Writer 200,–, Schachmeist 40,–, Buchungsjour. (o, Anl.) 60,–, Pers. Rec. Keeping + Pers. Rep. Gen. 80,–, Disk Man. 2 40,–, Parsec 30,–, Invad. 30,–.
Bücher: 99 Special 1 20,–, TI Basic/Exbs. f. A/F 15,–, Progr. f. den TI (Heigerm.) 15,–DM. Erich Niedergriese, Roenselstr. 9, 5800 Hagen 7, Tel: 02331/402220
P.S. Gibt's in Hagen keine TI-User? Meldet Euch!

Diskcontroller für Box inkl. Laufw. 490,– DM FP, 32 K Karte 300,– DM, alles 1 A Zustand, eben Original!! Tel: 06181/23399, ab 20 Uhr B. Pompe

---

Übersetz. E/A, über 400 S. + Probeexempl. von Zusatzinfo f. 49,80 DM. A Pack, Eickelerstr. 60, 4690 Herne 2

Verk. TI99/4A + Tape + 32 KRAM ext. (Atronic) + Speech Synth. + Exbasic + TI LOGO II + Moonsw. + Microsur. + Schach + Soccer + Min. Mis. + TI Joyst. + Quicks + 4 TI Kass. + über 80 Progr. + 5 Bücher + alle TI Rev. f. 650,– DM. Tel: 06134/4519

TI99/4A: Versandli. Disk 36,–, Lagerverw. Disk 36,–, DOSS Laufwerk + Contr. 650,–, RS232 ext. 2 Port. 180,–, 32 K + Centr. ex. 160,–, Exbas 100,–, Stat. 30,– Buchhaltung 50,–, Dateiverwl. + Analyse 40,–, Parsec 20,–, Diskmanag. 20,– DM. Tel: 06182/26186

Verk. TI + XBasic + Box + 32 K + RGB-Modulator + Rek. + Module: Statistik, Buchhaltung, Spielle + Kass. + Bücher: BASIC, XBASIC, E/A (dt.), TI-Intern, Masch. f. 700,– DM (auch einzeln). T. Klein, Cornelissenstr. 4, 4100 Duisburg 11, Tel: 0203/5981127

Verk. TI 99/4A 150,–, XBasic 100,–, 32 K ext. 150,–, Rek. + Kabel 60,–, Invaders 150,–, Parsec 15,– Pacman + Miner 2049 + Centipede + Espial 40,–, 2 Bücher 40,–, Turbo Ass. 60,–, Tunnels o. D. 50,–, 36 Mag. 50,–, kompl. 600,– DM Tel: 06421/44408

Suche günstig Seikosha SP 800 m. Kabel und dt. Anleit. f. Microsugeon u. TE II (auch Kopien mögl). Bitte melden bei: Thomas Nowak, Wangenheimstr. 36, 1000 Berlin 33, Tel: 030/891106, ab 14 Uhr. Nicht 29.7. - 17.8.)

Original TI an bestes Gebot über Mindestpreis Ex-Basic 110,–: Buch dazu 10,–, Ed/Ass 110,–, Minimem. 150,–, Statistik 30,–, Othello, Minusmiss., Alienadd., Rechnungste. je 15,–. Anruf. bei: H. Hägemann, Tel: 0531/82104, von 17 - 19 Uhr.

Verk. TI Komplettsystem!!! Konsole, P-Box m. orig. Laufw. 32K, RS232-Corcomp Controller umgebaut auf DOS 80 TI-Writer, E/A, Ex-Basic Ape-Soft, Literatur etc. Preis 1800,– VB. Tel: 042 30/1021, ab 18 Uhr

---

Verk. Hardware und Software für den TI 99/4A:
Module: Espial 30,–, Microsurg. 25,–, Dragon Mex 5,–, Hardware: neuw. Rec. 50,–, Rec.Kabel 10,–, Monitorkabel 10,–.
TI-Bücher: TI-Schaltpl. 5,–, „Spielen, lernen, arbeiten m. d. TI 99/4A" 25,–, TI-99 Tips + Tricks 25,–, TI 99/4A Farbe, Grafik, Ton u. Spiele von IWT 20,–, 21 LISTige Porgr. f. d. TI 99/4A 12,–, Spielprogr. selbst erstellen Teil 2 14,–, TI Basic Ext. Basic f. Anfäng. u. Fortgeschr. 25,–.
TI-Zeitschr.: TI-Rev.-Ausg.: 1 - 3/84, 1 - 9/85, 1 - 6/86 pro Stück 2,–, ab 10 Stück 1,50, TI Assembler Sepcial 9,–, TI Magazin 2 - 6/84 pro Stück 3,–, alle kompl. 10,–, TI 99/4A Programme, Sonderheft von CHIP 9,– DM. Bestellungen an: A. Schlaht, Große Str. 79, 2802 Ottersberg 1, Tel: 04205/1857, Mo - Fr 18.30 - 20.00 Uhr. Lieferung per NN oder Vorauskasse (V-Scheck) bzw. Euroscheck.

---

64 K Erweiterung für C 116/16 für 89,– DM bei Competer Soft – Fritz Peter – Konrad-Adenauer-Str. 201, 5130 Geilenkirchen, Tel: 02451/5061. Anwenderprogramme für Commodore C 16/116.

Verk. Joyst. Adapter 25,–, Module: Donkey Kong 40,–, Chess 50,–, Musicmaker 30,–, Adventuremod. auf Kass. 25,– DM. Tel: 09284/8338

DAS BERG'-SCHE SPIEL

Mit diesem Denkspiel kann man fast jeden zur Verzweiflung bringen. Alles, was man dazu braucht, ist ein TI-99/4A, ggf. mit Extended-Basic.
Der etwas seltsame Name dieses Programmes kommt daher, weil das Spiel eigentlich gar keinen richtigen Namen hat. Da aber der Mathelehrer, der mir dieses Spiel beigebracht hat, Herr Bergl hieß, habe ich es "Das Berg'sche Spiel" genannt. Der Zweitname "Das Kulispiel" kommt daher, weil die Gepäckkulis in Afrika dieses Spiel immer spielen, und der Verlierer die doppelte Last tragen muß (nach Hr. Berg!).
Aber nun zu den Spielregeln: In den vier Kästchen (A bis D) sind je eine bestimmte Anzahl von Hölzchen vorhanden. Sie und der TI nehmen abwechselnd aus je einem beliebigen Kästchen eine Anzahl von Hölzchen. Derjenige, der das letzte Hölzchen nimmt, hat verloren. Ist doch alles ganz einfach – sollte man meinen.
Der TI fragt nach Kästchen und Anzahl der Hölzchen, nimmt diese weg und überlegt sich innerhalb der nächsten drei Sekunden (!) seinen eigenen Zug. Ist im angegebenen Kästchen nur noch ein Hölzchen, so erübrigt sich für den TI die Frage nach der Anzahl.
Eine kleine Besonderheit hat das Programm aber noch: Wenn Sie der TI erst einmal so weit gebracht hat, daß Sie unweigerlich verlieren, wechselt die Schriftfarbe von Schwarz in Rot. Umgekehrt besagt eine Änderung nach Grün, daß Sie den TI ebenso unweigerlich besiegen können, wenn Sie keinen Denkfehler mehr machen (ein solcher hätte wieder die rote Schrift zur Folge, und Ihr TI macht garantiert keinen Fehler!). Eine Beschreibung der Variablen für dieses Programm habe ich nicht beigefügt, da man hieraus leicht die Regeln des Spiels ableiten könnte, und ich will den fleißen Programmtippern den Spaß nicht nehmen.

HINWEIS ZUM EINTIPPEN DES PROGRAMMES:

In den Zeilen 450 und 460 kommen sehr viele Controlzeichen vor. Sie müssen deshalb folgendermaßen getippt werden:

```
450 PRINT "SPACE/CTRL B/3*CTRL ,/CTRL F/
5*CTRL ,/CTRL F/6*CTRL ,/CTRL F/7*CTRL
,/CTRL C/2*SPACE/CTRL A/2*FCTN C/SPACE/
CTRL A/3*FCTN C/2*SPACE/CTRL A/4*FCTN
C/2*SPACE/CTRL A/5*FCTN C/2*SPACE/
CTRL A"
460 PRINT "SPACE/CTRL A/2*FCTN C/SPACE/
CTRL A/3*FCTN C/2*SPACE/CTRL A/4*FCTN
C/2*SPACE/CTRL A/5*FCTN C/2*SPACE/
CTRL A/2*SPACE/CTRL D/3*CTRL ,/CTRL
G/5*CTRL ,/CTRL G/6*CTRL ,/CTRL G/7*
CTRL ,/CTRL E"
```

Im Listing stehen einige seltsame Wörter. Diese sind kein Druckfehler, sondern enthalten umdefinierte Buchstaben und Zeichen, da es sonst mit dem COLOR-Befehl Knatsch gegeben hätte. Auch die Zeilen 3060 und 3430 mit dem BREAK sehen etwas seltsam aus. Sollte der TI jedoch wirklich einmal das Programm abbrechen, so ist noch irgendwo ein Tippfehler vorhanden, denn diese Zeilen dürfte er eigentlich niemals erreichen.

```
10 REM*********************
11 REM*DAS BERGSCHE SPIEL*
12 REM*                   *
13 REM*  Copyright  by    *
14 REM*   Sven Dyroff     *
15 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                   *
23 REM* Speicherbelegung *
24 REM*    8078 Bytes     *
25 REM*********************
26 REM
100 SG=2
110 DEMO=1
120 DIM POSITION(4,5),HELP(4
),KASTEN(4)
130 DATA 5,6,0,0,0,9,10,11,0
,0,15,16,17,18,0,22,23,24,25
,26
140 FOR I=1 TO 4
150 FOR J=1 TO 5
160 READ POSITION(I,J)
170 NEXT J
180 NEXT I
190 REM BILDAUFBAU
200 CALL CLEAR
210 CALL CHAR(89,"0000007824
382478")
220 CALL CHAR(95,"0000FFFF")
230 CALL CHAR(128,"000000FFF
F000000")
240 CALL CHAR(129,"181818181
8181818")
250 CALL CHAR(130,"0000001F1
F181818")
260 CALL CHAR(131,"000000F8F
8181818")
270 CALL CHAR(132,"1818181F1
F000000")
280 CALL CHAR(133,"181818F8F
8000000")
290 CALL CHAR(134,"000000FFF
F181818")
300 CALL CHAR(64,"0000003C40
40403C")
310 CALL CHAR(81,"0000007C40
78407C")
320 FOR Q=1 TO 4
330 KASTEN(Q)=Q+1
340 NEXT Q
350 CALL CHAR(135,"181818FFF
F000000")
360 CALL SCREEN(16)
370 CALL COLOR(13,16,16)
380 CALL COLOR(9,16,16)
390 CALL CHAR(96,"1818181818
181818")
400 IF DEMO=0 THEN 440
410 PRINT ".....DAS BERGSCHE
 SPIEL.....________________
______"
420 PRINT TAB(13);"Yzw.": :"
.....DAS KULISPIEL.....
.....________________": :
430 PRINT "...COMPUTERDEMONS
TRATION": :"....PRESS ENTER
TO BEGIN": : :
440 PRINT "...A....B.....C..
.....D....": :
450 PRINT "................
............''..'''..''''.
..''''''.....''..'''..''''.
..''''''...."
460 PRINT "..''..'''..''''.
..''''''.....''..'''..''''.
..''''''..................
.........."
470 RANDOMIZE
480 CALL COLOR(13,2,16)
490 CALL COLOR(9,5,16)
500 IF DEMO=0 THEN 620
510 REM SPIELEN 640 TO 2005
520 FOR Q=1 TO 50
530 CALL KEY(0,K,S)
540 IF K=13 THEN 570
550 NEXT Q
560 GOTO 1450
570 CALL SOUND(200,900,0)
580 CALL VCHAR(1,1,32,768)
590 DEMO=0
600 GEG=0
610 GOTO 370
620 DRUCK$="SCHWIERIGKEITSGR
AD ?  (1-3)"
630 ZEILE=6
640 SPALTE=3
650 GOSUB 3650
660 CALL KEY(0,K,S)
670 IF (K<49)+(K>51)THEN 660
680 CALL SOUND(200,900,0)
690 SG=K-48
700 CALL HCHAR(6,1,32,32)
710 GOTO 800
720 REM SPIELENDE
730 FOR Q=1 TO 150
740 CALL KEY(0,K,S)
750 IF K=13 THEN 570
760 NEXT Q
770 GEG=0
780 CALL VCHAR(1,1,32,768)
790 GOTO 320
800 FOR Q=1 TO 4
810 KASTEN(Q)=Q+1
820 NEXT Q
830 IF DEMO=1 THEN 1370
840 DRUCK$="WER SOLL ANFANGE
N ?"
850 ZEILE=5
860 SPALTE=4
870 GOSUB 3650
880 DRUCK$="DRUECKE I@h ODER
 ComputQr !"
890 ZEILE=7
900 SPALTE=1
910 GOSUB 3650
920 CALL KEY(0,K,S)
930 ON 3+(K=67)+(K=73)*2 GOT
O 940,960,920
```

```
940 CALL SOUND(200,900,0)
950 GOTO 1000
960 STRICH=2
970 KAST=1
980 GOSUB 3700
990 KASTEN(1)=1
1000 CALL HCHAR(5,1,32,32)
1010 CALL HCHAR(7,1,32,32)
1020 REM SPIELEN
1030 IF DEMO=1 THEN 510
1040 CALL HCHAR(1,1,32,224)
1050 DRUCK$="WELCHEN KASTEN
WILLST DU ?"
1060 ZEILE=5
1070 SPALTE=1
1080 GOSUB 3650
1090 CALL KEY(0,K,S)
1100 IF (K<65)+(K>68)THEN 10
90
1110 IF KASTEN(K-64)=0 THEN
1090
1120 KAST=K-64
1130 CALL HCHAR(5,1,32,32)
1140 CALL SOUND(200,900,0)
1150 DRUCK$="KAESTCHEN :  "&
CHR$(K)
1160 ZEILE=5
1170 SPALTE=1
1180 GOSUB 3650
1190 IF KASTEN(KAST)<>1 THEN
 1220
1200 K=49
1210 GOTO 1270
1220 DRUCK$="WIEVIELE STAEBE
 WILLST DU ?"
1230 ZEILE=7
1240 SPALTE=1
1250 GOSUB 3650
1260 CALL KEY(0,K,S)
1270 IF (K<49)+(K>KASTEN(KAS
T)+48)THEN 1260
1280 CALL HCHAR(7,1,32,32)
1290 CALL SOUND(200,900,0)
1300 DRUCK$="STAEBE :  "&CHR
$(K)
1310 ZEILE=7
1320 SPALTE=1
1330 GOSUB 3650
1340 GOSUB 3810
1350 CALL HCHAR(5,15,32)
1360 CALL HCHAR(7,12,32)
1370 REM
1380 REM
1390 REM
1400 DRUCK$="MEIN ZUG :"
1410 ZEILE=3
1420 SPALTE=9
1430 GOSUB 3650
1440 CALL HCHAR(4,10,95,10)
1450 REM
1460 AN$=""
1470 FOR Q=1 TO 4
1480 IF KASTEN(Q)=0 THEN 150
0
1490 AN$=AN$&STR$(Q)
1500 NEXT Q
1510 IF LEN(AN$)=0 THEN 1580
1520 REM
1530 IF LEN(AN$)<>1 THEN 191
0
1540 IF KASTEN(VAL(AN$))=1 T
HEN 1820
1550 KAST=VAL(AN$)
1560 K=KASTEN(VAL(AN$))+47
1570 GOSUB 3810
1580 IF DEMO=1 THEN 720
1590 CALL HCHAR(1,1,32,224)
1600 DRUCK$="SIE HABEN VERLO
REN."
1610 ZEILE=5
1620 SPALTE=3
1630 GOSUB 3650
1640 DRUCK$="NOCH EIN SPIEL
?  (J/N)"
1650 ZEILE=7
1660 SPALTE=3
1670 GOSUB 3650
1680 FOR Q=1 TO 250
1690 CALL KEY(0,K,S)
1700 ON 3+(K=74)+(K=78)*2 GO
TO 1740,1760,1710
1710 NEXT Q
1720 DEMO=1
1730 GOTO 1770
1740 CALL CLEAR
1750 END
1760 CALL SOUND(200,900,0)
1770 FOR Q=1 TO 14
1780 CALL COLOR(Q,2,16)
1790 NEXT Q
1800 SYM1=0
1810 GOTO 190
1820 REM
1830 IF DEMO=1 THEN 720
1840 CALL HCHAR(1,1,32,224)
1850 DRUCK$="SIE HABEN GEWON
NEN."
1860 ZEILE=5
1870 SPALTE=3
1880 GOSUB 3650
1890 GOTO 1640
1900 REM
1910 AS$=""
1920 FOR I=1 TO LEN(AN$)
1930 FOR J=I TO LEN(AN$)
1940 IF I=J THEN 1960
1950 IF KASTEN(VAL(SEG$(AN$,
I,1)))=KASTEN(VAL(SEG$(AN$,J
,1)))THEN 2000
1960 NEXT J
1970 AS$=AS$&SEG$(AN$,I,1)
1980 NEXT I
1990 GOTO 2680
2000 FOR Q=I+1 TO LEN(AN$)
2010 IF Q=J THEN 2030
2020 AS$=AS$&SEG$(AN$,Q,1)
```

LISTINGS

```
2030 NEXT Q
2040 REM
2050 IF KASTEN(VAL(SEG$(AN$,
J,1)))=1 THEN 2080
2060 SYM1=0
2070 GOTO 2090
2080 SYM1=1
2090 REM
2100 IF LEN(AS$)<>0 THEN 223
0
2110 IF SYM1<>1 THEN 2170
2120 KAST=VAL(SEG$(AN$,1,1))
2130 GOSUB 3960
2140 K=49
2150 GOSUB 3810
2160 GOTO 1580
2170 GOSUB 4030
2180 KAST=VAL(SEG$(AN$,1,1))
2190 K=INT(RND*KASTEN(KAST)+
49)
2200 GOSUB 3810
2210 GOTO 1020
2220 REM
2230 IF LEN(AS$)<>1 THEN 238
0
2240 IF SYM1=1 THEN 2310
2250 GOSUB 3960
2260 KAST=VAL(AS$)
2270 K=KASTEN(KAST)+48
2280 GOSUB 3810
2290 GOTO 1020
2300 REM
2310 IF KASTEN(VAL(AS$))<>1
THEN 2330
2320 GOTO 1830
2330 GOSUB 3960
2340 KAST=VAL(AS$)
2350 K=KASTEN(KAST)+47
2360 GOSUB 3810
2370 GOTO 1580
2380 REM
2390 IF KASTEN(VAL(SEG$(AS$,
1,1)))<>KASTEN(VAL(SEG$(AS$,
2,1)))THEN 2450
2400 GOSUB 4030
2410 KAST=VAL(SEG$(AN$,INT(R
ND*4)+1,1))
2420 K=INT(RND*KASTEN(KAST))
+49
2430 GOSUB 3810
2440 GOTO 1020
2450 IF SYM1=1 THEN 2570
2460 GOSUB 2510
2470 GOSUB 3960
2480 K=K+48
2490 GOSUB 3810
2500 GOTO 1020
2510 I=VAL(SEG$(AS$,1,1))
2520 J=VAL(SEG$(AS$,2,1))
2530 Q=KASTEN(I)>KASTEN(J)
2540 KAST=-I*Q+(Q+1)*J
2550 K=ABS(KASTEN(I)-KASTEN(
J))
2560 RETURN
2570 REM
2580 GOSUB 2510
2590 IF KASTEN(KAST)-K<>1 TH
EN 2640
2600 K=K+49
2610 GOSUB 3960
2620 GOSUB 3810
2630 GOTO 1580
2640 K=K+48
2650 GOSUB 3960
2660 GOSUB 3810
2670 GOTO 1020
2680 REM
2690 HELP(0)=LEN(AN$)
2700 FOR Q=1 TO HELP(0)
2710 HELP(Q)=VAL(SEG$(AN$,Q,
1))
2720 NEXT Q
2730 IF HELP(0)<>2 THEN 2870
2740 FOR Q=1 TO 2
2750 IF KASTEN(HELP(Q))<>1 T
HEN 2810
2760 KAST=HELP((Q-1.5)*-1+1.
5)
2770 K=KASTEN(KAST)+48
2780 GOSUB 3960
2790 GOSUB 3810
2800 GOTO 1580
2810 NEXT Q
2820 GOSUB 2510
2830 K=K+48
2840 GOSUB 3960
2850 GOSUB 3810
2860 GOTO 1020
2870 ON SG GOTO 2890,2930,33
00
2880 GOSUB 4030
2890 KAST=HELP(INT(RND*HELP(
0)+1))
2900 K=INT(RND*KASTEN(KAST)+
49)
2910 GOSUB 3810
2920 GOTO 1020
2930 @@@=0
2940 IF HELP(0)=4 THEN 3130
2950 FOR I=1 TO 3
2960 FOR J=1 TO 3
2970 IF KASTEN(HELP(J))=I TH
EN 3010
2980 NEXT J
2990 IF @@@<>0 THEN 2890
3000 @@@=I
3010 NEXT I
3020 IF @@@=0 THEN 2880
3030 FOR Q=1 TO 3
3040 IF KASTEN(HELP(Q))>3.5
THEN 3070
3050 NEXT Q
3060 BREAK
3070 KAST=HELP(Q)
3080 IF KASTEN(KAST)-@@@<1 T
HEN 2890
```

```
3090 K=KASTEN(KAST)-@@@+48
3100 GOSUB 3960
3110 GOSUB 3810
3120 GOTO 1020
3130 FOR I=1 TO 3
3140 FOR J=1 TO 4
3150 IF KASTEN(HELP(J))=I TH
EN 3180
3160 NEXT J
3170 GOTO 2890
3180 FOR Q=J+1 TO 4
3190 IF KASTEN(HELP(Q))=I TH
EN 3250
3200 NEXT Q
3210 NEXT I
3220 FOR Q=1 TO 4
3230 IF KASTEN(HELP(Q))>3.5
THEN 3250
3240 NEXT Q
3250 KAST=HELP(Q)
3260 K=KASTEN(KAST)+48
3270 GOSUB 3960
3280 GOSUB 3810
3290 GOTO 1020
3300 @@@=0
3310 IF HELP(0)=4 THEN 3490
3320 FOR I=3 TO 5
3330 FOR J=1 TO 3
3340 IF KASTEN(HELP(J))=I+2*
(I=3)THEN 3380
3350 NEXT J
3360 IF @@@<>0 THEN 2930
3370 @@@=I+2*(I=3)
3380 NEXT I
3390 IF @@@=0 THEN 2890
3400 FOR Q=1 TO 3
3410 IF ABS(KASTEN(HELP(Q))-
2.5)=0.5 THEN 3440
3420 NEXT Q
3430 BREAK
3440 KAST=HELP(Q)
3450 IF KASTEN(KAST)-@@@<1 T
HEN 2930
3460 K=KASTEN(KAST)-@@@+48
3470 GOSUB 3810
3480 GOTO 1020
3490 FOR I=3 TO 5
3500 FOR J=1 TO 4
3510 IF KASTEN(HELP(J))=I+2*
(I=3)THEN 3540
3520 NEXT J
3530 GOTO 2930
3540 FOR Q=J+1 TO 4
3550 IF KASTEN(HELP(Q))=I+2*
(I=3)THEN 3610
3560 NEXT Q
3570 NEXT I
3580 FOR Q=1 TO 4
3590 IF ABS(KASTEN(HELP(Q))-
2.5)=0.5 THEN 3610
3600 NEXT Q
3610 KAST=HELP(Q)
3620 K=KASTEN(KAST)+48
3630 GOSUB 3810
3640 GOTO 1020
3650 REM PRINTROUTINE
3660 FOR Q=1 TO LEN(DRUCK$)
3670 CALL HCHAR(ZEILE,SPALTE
+Q,ASC(SEG$(DRUCK$,Q,1)))
3680 NEXT Q
3690 RETURN
3700 REM STRICH WEGNEHMEN
3710 FOR Q=1 TO 3
3720 CALL VCHAR(19,POSITION(
KAST,STRICH),96,4)
3730 CALL SOUND(200,250,0)
3740 CALL SOUND(1,40000,30)
3750 CALL VCHAR(19,POSITION(
KAST,STRICH),32,4)
3760 CALL SOUND(200,250,0)
3770 CALL SOUND(1,40000,30)
3780 NEXT Q
3790 CALL SOUND(300,135,0)
3800 RETURN
3810 IF DEMO=0 THEN 3880
3820 IF GEG=0 THEN 3840
3830 GEG=-1
3840 GEG=GEG+1
3850 CALL HCHAR(24,30-GEG*28
,KAST+64)
3860 CALL HCHAR(24,31-GEG*28
,K)
3870 GOTO 3900
3880 CALL HCHAR(5,15,KAST+64
)
3890 CALL HCHAR(7,12,K)
3900 FOR J=1 TO K-48
3910 STRICH=KASTEN(KAST)
3920 GOSUB 3700
3930 KASTEN(KAST)=KASTEN(KAS
T)-1
3940 NEXT J
3950 RETURN
3960 REM COLOR ROT
3970 IF DEMO=1 THEN 4020
3980 FOR Q=1 TO 14
3990 CALL COLOR(Q,7,16)
4000 NEXT Q
4010 CALL COLOR(9,5,16)
4020 RETURN
4030 REM COLOR GRUEN
4040 IF DEMO=1 THEN 4020
4050 FOR Q=1 TO 14
4060 CALL COLOR(Q,13,16)
4070 NEXT Q
4080 GOTO 4010
```

LISTINGS

HUNGER

Kennwort „Hungerhilfe Afrika"

Menschen in Not brauchen Hilfe: zuverlässig, schnell, wirksam. Die beiden kirchlichen Hilfswerke nehmen ihren Auftrag ernst.

Deutscher Caritasverband Postgiro Karlsruhe 202
Diakonisches Werk Postgiro Stuttgart 502
und viele Banken u. Sparkassen

LISTING

EAN-CODER 13

Das Programm zur Erstellung von scannerlesbaren(!) EAN-Codes, Ausdruck mit Printer und Speicherung auf Cassette.
Bei diesem Programm wurden alle Regeln des 13-stelligen EAN-Codes (A-/B-/C-Code, 1-3-1 Prüfziffercode) bis hin zur Ermittlung der Prüfziffer, bzw. deren Kontrolle und der visuell lesbaren OCR-B Klarschriftzeile unter dem Code. Weiterhin druckt das Programm bis zu 4 interne Kurzbezeichnungen neben den Code, die ebenfalls abgespeichert werden können.

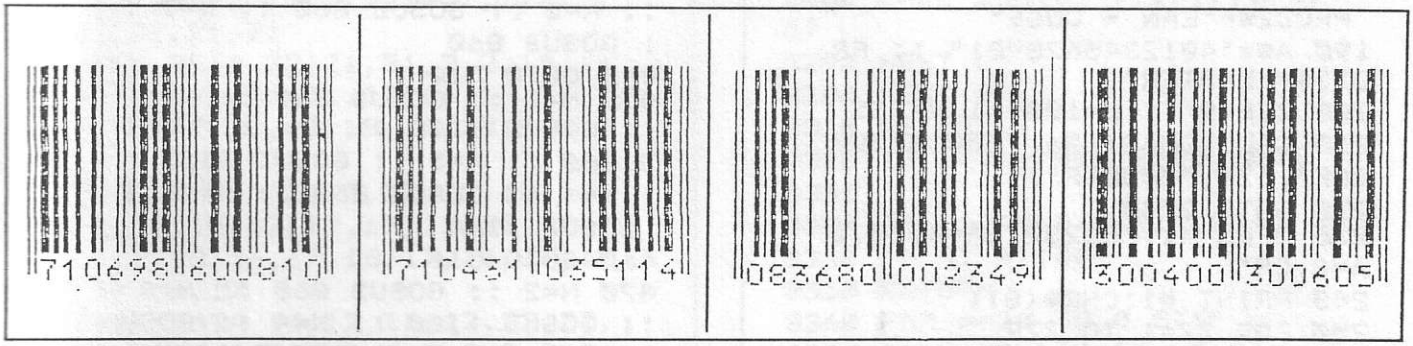
--- Achtung Copyright by Datecontrol ---

Mit diesem Programm können Sie jeden beliebigen 13stelligen EAN-Code entwickeln, die Prüfziffer kontrollieren bzw. erstellen und abspeichern bzw. ausdrucken lassen.

Als Printer wurde ein Seikosha GP 100 A benutzt, jedoch dürfte es ohne große Programmänderungen in den Druckzeilen möglich sein, jeden anderen Matrixprinter anzusteuern.
Auf die o.e. Punkte 1,3,4 braucht man sicher nicht gesondert einzugehen. Wählen Sie hingegen Punkt 2, gelangen Sie ins Hauptprogramm.

DAS HAUPTPROGRAMM

Dargestellt ist die In-/Output-Maske mit einem Muster-EAN-Code. In der letzten Zeile sind die aktiven Tasten angegeben.
Wählen Sie "N" für Neueingabe, "P" für Ausdruck und "M" für Menü.
Drücken Sie jetzt <ENTER> bzw. "N" und <ENTER>, haben Sie die Möglichkeit, einen Code einzugeben.



Die EAN-Code-Ausdrucke dieses Programms sind scannerlesbar, d.h., sie können mit Lesestiften an Scannerkassen in Supermärkten oder an Lesesystemen gelesen werden. Deshalb sei hier besonders darauf hingewiesen, daß eine solche oder anderweitige gewerbliche bzw. nicht-private Nutzung strafrechtlich geahndet wird. Außerdem gelten natürlich auch die üblichen Urheberrechte.
Auf die Einzelheiten des EAN-Codes einzugehen, dürfte hier nicht angebracht sein, da es den Rahmen des Heftes sprengen würde.
Zum Verständnis des Programmes ist dies auch nicht unbedingt notwendig. Eine kurze Darstellung der Zusammensetzung des Codes finden Sie jedoch nachfolgend:

1 = Länderkenner, z.B. 40 für Deutschland
2 = bundeseinheitl. Betriebsnummer BBN
3 = individuelle Artikelnummer des Herstellers
4 = Prüfziffer

DAS PROGRAMM

Nach dem Einlesen des Extended-Basic-Programms in Ihren TI 99/4A starten Sie das Programm mit <RUN> (bzw. Autostart mit < RUN "CS1">)
Dargestellt ist das Hauptmenü.
Wählen Sie jetzt zwischen:

01 – DATEN EINLESEN
02 – HAUPTPROGRAMM/DATEN AUSDRUCKEN
03 – DATEN ABSPEICHERN
04 – PROGRAMMENDE

A – Eingabe von 13 Stellen mit Kontrolle des Codes:
Geben Sie jetzt eine 13stellige Ziffernfolge (EAN-Code) ein.
Der Computer prüft, ob der Code korrekt ist und der Prüfziffer entspricht. Ist dies nicht der Fall, erscheint neben der Länderbezeichnung ein "NOT OK" und die Frage "GO ON?" (weiter?).
Antworten Sie hier mit "Y" (Yes/Ja) können Sie mit dem unkorrekten Code im Programm fortfahren.
Zur erneuten Eingabe bzw. Korrektur drücken Sie "N" (Nein) und geben den neuen Code ein. Ist der Code nun in Ordnung, erscheint ">>>OK" sowie die Frage "GO ON?". Drücken Sie "Y" für JA/Yes, geht's weiter im Programm. Die 4 folgenden Zeilen, die jetzt abgefragt werden, werden auf Wunsch neben dem EAN-Code ausgedruckt. Wünschen Sie keinen Ausdruck einzelner Zeilen, geben Sie Leerzeichen ein. Ansonsten können Sie hier Angaben über Preis, Gewicht, Firma etc. machen.
In der Maske steht "FRM" für Firma, "CAT" für Categorie und "ART" für Artikel.
B – Eingabe von 12 Stellen und "P" zur Ermittlung der Prüfziffer:
Neben der Eingabe des gesamten 13stelligen Codes haben Sie die Möglichkeit, die Prüfziffer vom Com-

Bitte lesen Sie weiter auf Seite 121

LISTINGS

```
10 ! ************************
11 ! *                      *
12 ! *    EAN-CODER 13      *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *                      *
16 ! *  Thomas Jerusalem    *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *  Kassettenrekorder   *
23 ! * Drucker(Seikosha-    *
24 ! * GP 100A)             *
25 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *    10946 Bytes       *
28 ! *                      *
29 ! ************************
30 !
160 ON BREAK NEXT :: ON ERRO
R 2040
170 T1$="FRM" :: T2$="CAT" :
: T3$="ART" :: T4$="ART"
180 NAME$="DATECONTROL, NEUW
IED" :: PROD1$="SOFTWARE" ::
 PROD2$="EAN - CODE"
190 A$="401234567891" :: PR
OD3$="MASTER"
200 ZZE=9 :: L=128 :: V=255
210 CALL CHAR(126,"000000000
0FF",124,"0000FF")
220 GOTO 2040
230 OPEN #1:"RS232.BA=4800.D
A=8.CR"
240 PRINT #1:CHR$(8);
250 FOR ZZ=1 TO ZZE
260 PRINT #1:CHR$(28);CHR$(1
3);CHR$(L);CHR$(255);CHR$(L)
;CHR$(255);
270 B$=SEG$(A$,1,1)
280 IF B$="0" THEN 310 ELSE
IF B$="1" THEN 330 ELSE IF B
$="2" THEN 350 ELSE IF B$="3
" THEN 370
290 IF B$="4" THEN 390 ELSE
IF B$="5" THEN 410 ELSE IF B
$="6" THEN 430 ELSE IF B$="7
" THEN 450
300 IF B$="8" THEN 470 ELSE
IF B$="9" THEN 490 ELSE 120
310 N=2 :: GOSUB 860 :: N=3
:: GOSUB 860 :: N=4 :: GOSUB
 860 :: N=5 :: GOSUB 860 ::
N=6 :: GOSUB 860 :: N=7 :: G
OSUB 860
320 GOTO 510
330 N=2 :: GOSUB 860 :: N=3
:: GOSUB 860 :: N=4 :: GOSUB
 1100 :: N=5 :: GOSUB 860 ::
 N=6 :: GOSUB 1100 :: N=7 ::
 GOSUB 1100
340 GOTO 510
350 N=2 :: GOSUB 860 :: N=3
:: GOSUB 860 :: N=4 :: GOSUB
 1100 :: N=5 :: GOSUB 1100 :
: N=6 :: GOSUB 860 :: N=7 ::
 GOSUB 1100
360 GOTO 510
370 N=2 :: GOSUB 860 :: N=3
:: GOSUB 860 :: N=4 :: GOSUB
 1100 :: N=5 :: GOSUB 1100 :
: N=6 :: GOSUB 1100 :: N=7 :
: GOSUB 860
380 GOTO 510
390 N=2 :: GOSUB 860 :: N=3
:: GOSUB 1100 :: N=4 :: GOSU
B 860 :: N=5 :: GOSUB 860 ::
 N=6 :: GOSUB 1100 :: N=7 ::
 GOSUB 1100
400 GOTO 510
410 N=2 :: GOSUB 860 :: N=3
:: GOSUB 1100 :: N=4 :: GOSU
B 1100 :: N=5 :: GOSUB 860 :
: N=6 :: GOSUB 860 :: N=7 ::
 GOSUB 1100
420 GOTO 510
430 N=2 :: GOSUB 860 :: N=3
:: GOSUB 1100 :: N=4 :: GOSU
B 1100 :: N=5 :: GOSUB 1100
:: N=6 :: GOSUB 860 :: N=7 :
: GOSUB 860
440 GOTO 510
450 N=2 :: GOSUB 860 :: N=3
:: GOSUB 1100 :: N=4 :: GOSU
B 860 :: N=5 :: GOSUB 1100 :
: N=6 :: GOSUB 860 :: N=7 ::
 GOSUB 1100
460 GOTO 510
470 N=2 :: GOSUB 860 :: N=3
:: GOSUB 1100 :: N=4 :: GOSU
B 860 :: N=5 :: GOSUB 1100 :
: N=6 :: GOSUB 1100 :: N=7 :
: GOSUB 860
480 GOTO 510
490 N=2 :: GOSUB 860 :: N=3
:: GOSUB 1100 :: N=4 :: GOSU
B 1100 :: N=5 :: GOSUB 860 :
: N=6 :: GOSUB 1100 :: N=7 :
: GOSUB 860
500 GOTO 510
510 PRINT #1:CHR$(L);CHR$(25
5);CHR$(L);CHR$(255);CHR$(L)
;
520 N=8 :: GOSUB 620 :: N=9
:: GOSUB 620 :: N=10 :: GOSU
B 620 :: N=11 :: GOSUB 620 :
: N=12 :: GOSUB 620 :: N=13
:: GOSUB 620 ::
530 PRINT #1:CHR$(255);CHR$(
L);CHR$(255):: !;CHR$(13);CH
R$(10)
540 IF ZZ=1 THEN GOSUB 1860
:: GOTO 590
550 IF ZZ=3 THEN GOSUB 1880
:: GOTO 590
560 IF ZZ=5 THEN GOSUB 1900
:: GOTO 590
```

```
570 IF ZZ=7 THEN GOSUB 1920
 :: GOTO 590
580 IF ZZ=8 OR ZZ=9 THEN GOSUB 61
960
590 IF ZZ=ZZE-1 THEN V=159
600 NEXT ZZ :: V=255 :: GOTO
 1340
610 ! C 6-1
620 XX$=SEG$(A$,N,1)
630 IF XX$="0" THEN 650 ELSE
 IF XX$="1" THEN 670 ELSE IF
 XX$="2" THEN 690 ELSE IF XX
$="3" THEN 710 ELSE IF XX$="
4" THEN 730
640 IF XX$="5" THEN 750 ELSE
 IF XX$="6" THEN 770 ELSE IF
 XX$="7" THEN 790 ELSE IF XX
$="8" THEN 810 ELSE IF XX$="
9" THEN 830
650 PRINT #1:CHR$(V);CHR$(V)
;CHR$(V);CHR$(L);CHR$(L);CHR
$(V);CHR$(L);
660 RETURN
670 PRINT #1:CHR$(V);CHR$(V)
;CHR$(L);CHR$(L);CHR$(V);CHR
$(V);CHR$(L);
680 RETURN
690 PRINT #1:CHR$(V);CHR$(V)
;CHR$(L);CHR$(V);CHR$(V);CHR
$(L);CHR$(L);
700 RETURN
710 PRINT #1:CHR$(V);CHR$(L)
;CHR$(L);CHR$(L);CHR$(L);CHR
$(V);CHR$(L);
720 RETURN
730 PRINT #1:CHR$(V);CHR$(L)
;CHR$(V);CHR$(V);CHR$(V);CHR
$(L);CHR$(L);
740 RETURN
750 PRINT #1:CHR$(V);CHR$(L)
;CHR$(L);CHR$(V);CHR$(V);CHR
$(V);CHR$(L);
760 RETURN
770 PRINT #1:CHR$(V);CHR$(L)
;CHR$(V);CHR$(L);CHR$(L);CHR
$(L);CHR$(L);
780 RETURN
790 PRINT #1:CHR$(V);CHR$(L)
;CHR$(L);CHR$(L);CHR$(V);CHR
$(L);CHR$(L);
800 RETURN
810 PRINT #1:CHR$(V);CHR$(L)
;CHR$(L);CHR$(V);CHR$(L);CHR
$(L);CHR$(L);
820 RETURN
830 PRINT #1:CHR$(V);CHR$(V)
;CHR$(V);CHR$(L);CHR$(V);CHR
$(L);CHR$(L);
840 RETURN
850 END
860 ! A 12-7
870 XX$=SEG$(A$,N,1)
880 IF XX$="0" THEN 900 ELSE
 IF XX$="1" THEN 920 ELSE IF
 XX$="2" THEN 940 ELSE IF XX
$="3" THEN 960 ELSE IF XX$="
4" THEN 980
890 IF XX$="5" THEN 1000 ELS
E IF XX$="6" THEN 1020 ELSE
IF XX$="7" THEN 1040 ELSE IF
 XX$="8" THEN 1060 ELSE IF X
X$="9" THEN 1080
900 PRINT #1:CHR$(L);CHR$(L)
;CHR$(L);CHR$(V);CHR$(V);CHR
$(L);CHR$(V);
910 RETURN
920 PRINT #1:CHR$(L);CHR$(L)
;CHR$(V);CHR$(V);CHR$(L);CHR
$(L);CHR$(V);
930 RETURN
940 PRINT #1:CHR$(L);CHR$(L)
;CHR$(V);CHR$(L);CHR$(L);CHR
$(V);CHR$(V);
950 RETURN
960 PRINT #1:CHR$(L);CHR$(V)
;CHR$(V);CHR$(V);CHR$(V);CHR
$(L);CHR$(V);
970 RETURN
980 PRINT #1:CHR$(L);CHR$(V)
;CHR$(L);CHR$(L);CHR$(L);CHR
$(V);CHR$(V);
990 RETURN
1000 PRINT #1:CHR$(L);CHR$(V
);CHR$(V);CHR$(L);CHR$(L);CH
R$(L);CHR$(V);
1010 RETURN
1020 PRINT #1:CHR$(L);CHR$(V
);CHR$(L);CHR$(V);CHR$(V);CH
R$(V);CHR$(V);
1030 RETURN
1040 PRINT #1:CHR$(L);CHR$(V
);CHR$(V);CHR$(V);CHR$(L);CH
R$(V);CHR$(V);
1050 RETURN
1060 PRINT #1:CHR$(L);CHR$(V
);CHR$(V);CHR$(L);CHR$(V);CH
R$(V);CHR$(V);
1070 RETURN
1080 PRINT #1:CHR$(L);CHR$(L
);CHR$(L);CHR$(V);CHR$(L);CH
R$(V);CHR$(V);
1090 RETURN
1100 ! B 12-7
1110 XX$=SEG$(A$,N,1)
1120 IF XX$="0" THEN 1140 EL
SE IF XX$="1" THEN 1160 ELSE
 IF XX$="2" THEN 1180 ELSE I
F XX$="3" THEN 1200 ELSE IF
XX$="4" THEN 1220
1130 IF XX$="5" THEN 1240 EL
SE IF XX$="6" THEN 1260 ELSE
 IF XX$="7" THEN 1280 ELSE I
F XX$="8" THEN 1300 ELSE IF
XX$="9" THEN 1320
1140 PRINT #1:CHR$(L);CHR$(V
);CHR$(L);CHR$(L);CHR$(V);CH
```

```
R$(V);CHR$(V);
1150 RETURN
1160 PRINT #1:CHR$(L);CHR$(V
);CHR$(V);CHR$(L);CHR$(L);CH
R$(V);CHR$(V);
1170 RETURN
1180 PRINT #1:CHR$(L);CHR$(L
);CHR$(V);CHR$(V);CHR$(L);CH
R$(V);CHR$(V);
1190 RETURN
1200 PRINT #1:CHR$(L);CHR$(V
);CHR$(L);CHR$(L);CHR$(L);CH
R$(L);CHR$(V);
1210 RETURN
1220 PRINT #1:CHR$(L);CHR$(L
);CHR$(V);CHR$(V);CHR$(V);CH
R$(L);CHR$(V);
1230 RETURN
1240 PRINT #1:CHR$(L);CHR$(V
);CHR$(V);CHR$(V);CHR$(L);CH
R$(L);CHR$(V);
1250 RETURN
1260 PRINT #1:CHR$(L);CHR$(L
);CHR$(L);CHR$(L);CHR$(V);CH
R$(L);CHR$(V);
1270 RETURN
1280 PRINT #1:CHR$(L);CHR$(L
);CHR$(V);CHR$(L);CHR$(L);CH
R$(L);CHR$(V);
1290 RETURN
1300 PRINT #1:CHR$(L);CHR$(L
);CHR$(L);CHR$(V);CHR$(L);CH
R$(L);CHR$(V);
1310 RETURN
1320 PRINT #1:CHR$(L);CHR$(L
);CHR$(V);CHR$(L);CHR$(V);CH
R$(V);CHR$(V);
1330 RETURN
1340 PRINT #1:CHR$(28);CHR$(
13);CHR$(L);CHR$(159);CHR$(L
);CHR$(159);CHR$(28);CHR$(43
);CHR$(L);CHR$(159);CHR$(L);
CHR$(159);
1350 PRINT #1:CHR$(28);CHR$(
43);CHR$(L);CHR$(159);CHR$(L
);CHR$(159);CHR$(13)
1360 CLOSE #1
1370 OPEN #1:"RS232.BA=4800.
DA=8.CR"
1380 FOR NN=1 TO 13
1390 MM$=SEG$(A$,NN,1)
1400 IF MM$="0" THEN 1420 EL
SE IF MM$="1" THEN 1450 ELSE
 IF MM$="2" THEN 1480 ELSE I
F MM$="3" THEN 1510 ELSE IF
MM$="4" THEN 1540
1410 IF MM$="5" THEN 1570 EL
SE IF MM$="6" THEN 1600 ELSE
 IF MM$="7" THEN 1630 ELSE I
F MM$="8" THEN 1660 ELSE IF
MM$="9" THEN 1690
1420 GOSUB 1800
1430 PRINT #1:CHR$(128);CHR$
(190);CHR$(193);CHR$(193);CH
R$(193);CHR$(190);CHR$(128);
1440 GOTO 1720
1450 GOSUB 1800
1460 PRINT #1:CHR$(128);CHR$
(132);CHR$(130);CHR$(255);CH
R$(128);CHR$(128);CHR$(128);
1470 GOTO 1720
1480 GOSUB 1800
1490 PRINT #1:CHR$(128);CHR$
(226);CHR$(209);CHR$(209);CH
R$(201);CHR$(198);CHR$(128);
1500 GOTO 1720
1510 GOSUB 1800
1520 PRINT #1:CHR$(128);CHR$
(161);CHR$(201);CHR$(205);CH
R$(203);CHR$(177);CHR$(128);
1530 GOTO 1720
1540 GOSUB 1800
1550 PRINT #1:CHR$(128);CHR$
(152);CHR$(148);CHR$(146);CH
R$(249);CHR$(144);CHR$(128);
1560 GOTO 1720
1570 GOSUB 1800
1580 PRINT #1:CHR$(128);CHR$
(199);CHR$(197);CHR$(197);CH
R$(197);CHR$(184);CHR$(128);
1590 GOTO 1720
1600 GOSUB 1800
1610 PRINT #1:CHR$(128);CHR$
(184);CHR$(204);CHR$(202);CH
R$(201);CHR$(176);CHR$(128);
1620 GOTO 1720
1630 GOSUB 1800
1640 PRINT #1:CHR$(128);CHR$
(129);CHR$(129);CHR$(249);CH
R$(133);CHR$(131);CHR$(128);
1650 GOTO 1720
1660 GOSUB 1800
1670 PRINT #1:CHR$(128);CHR$
(182);CHR$(201);CHR$(201);CH
R$(201);CHR$(182);CHR$(128);
1680 GOTO 1720
1690 GOSUB 1800
1700 PRINT #1:CHR$(128);CHR$
(134);CHR$(201);CHR$(169);CH
R$(153);CHR$(142);CHR$(128);
1710 GOTO 1720
1720 NEXT NN
1730 GOSUB 1940
1740 PRINT #1:CHR$(10);CHR$(
13)
1750 CLOSE #1
1760 OPEN #1:"RS232.BA=4800.
DA=8"
1770 PRINT #1: : : :
1780 CLOSE #1
1790 DISPLAY AT(24,1):"" ::
GOTO 2990
1800 ! SPACE
1810 IF NN=2 THEN 1820 ELSE
IF NN=8 THEN 1840 ELSE RETUR
N
```

```
1820 PRINT #1:CHR$(28);CHR$(
9);CHR$(128);
1830 RETURN
1840 PRINT #1:CHR$(28);CHR$(
5);CHR$(128);
1850 RETURN
1860 PRINT #1:CHR$(15);"
 ";NAME$;CHR$(8);CHR$(13);CH
R$(10)
1870 RETURN
1880 PRINT #1:CHR$(15);"
 ";PROD1$;CHR$(8);CHR$(13);C
HR$(10)
1890 RETURN
1900 PRINT #1:CHR$(15);"
 ";PROD2$;CHR$(8);CHR$(13);C
HR$(10)
1910 RETURN
1920 PRINT #1:CHR$(15);"
 ";PROD3$;CHR$(8);CHR$(13);C
HR$(10)
1930 RETURN
1940 PRINT #1:CHR$(28);CHR$(
3);CHR$(128);CHR$(15);"
";"by DATECONTROL";CHR$(8);C
HR$(13);CHR$(10)
1950 RETURN
1960 PRINT #1:CHR$(13);CHR$(
10)
1970 RETURN
1980 CALL HCHAR(4,2,129):: C
ALL HCHAR(5,2,91):: CALL HCH
AR(6,2,123):: CALL VCHAR(7,2
,91,3):: CALL HCHAR(10,2,123
):: CALL VCHAR(11,2,91,3)
1990 CALL HCHAR(14,2,123)::
CALL VCHAR(15,2,91,3):: CALL
 HCHAR(18,2,123):: CALL VCHA
R(19,2,91,3):: CALL HCHAR(22
,2,130)
2000 CALL HCHAR(4,22,133)::
CALL HCHAR(5,22,134):: CALL
HCHAR(6,22,135)
2010 CALL HCHAR(4,31,131)::
CALL HCHAR(5,31,93):: CALL H
CHAR(6,31,125):: CALL VCHAR(
7,31,93,3):: CALL HCHAR(10,3
1,125):: CALL VCHAR(11,31,93
,3)
2020 CALL HCHAR(14,31,125)::
 CALL VCHAR(15,31,93,3):: CA
LL HCHAR(18,31,125):: CALL V
CHAR(19,31,93,3):: CALL HCHA
R(22,31,132)
2030 RETURN
2040 CALL CLEAR :: CALL SCRE
EN(2):: FOR I=0 TO 14 :: CAL
L COLOR(I,7,1):: NEXT I
2050 DISPLAY AT(1,1):"EAN -
CODER..........EAN 13"
2060 DISPLAY AT(8,1):"0001..
.LOAD DATA.....CS1": :"0
002   PROGRAM/PRINT...RS232"
: :"0003...SAVE DATA.......
.CS1"
2070 DISPLAY AT(14,1):"0004.
..END OF PROGRAMM.....E" ::
DISPLAY AT(24,1):"...all rig
hts by DateControl"
2080 CALL KEY(0,K,S):: IF S=
0 THEN 2080 ELSE IF K=49 THE
N 2160 ELSE IF K=50 THEN 210
0 ELSE IF K=51 THEN 2230
2090 IF K=52 THEN 2120 ELSE
2080
2100 CALL CLEAR :: DISPLAY A
T(1,1):"EAN - CODER 13...Dat
eControl"
2110 GOTO 2840
2120 CALL CLEAR :: DISPLAY A
T(7,1):".....EAN - CODER":
: :".....all rights by": :"
.....DateControl": :".....
MONDWEG 0009":".....D 5450
NEUWIED 1"
2130 DISPLAY AT(15,1):".....
.WEST - GERMANY": :".....#
02631 / 26364"
2140 DISPLAY AT(24,16):"E/EN
D  G/O ON"
2150 CALL KEY(0,K,S):: IF S=
0 THEN 2150 ELSE IF K=69 THE
N END ELSE IF K=71 THEN 2040
 ELSE END
2160 CALL CLEAR
2170 OPEN #3:"CS1",INTERNAL,
INPUT ,FIXED 192
2180 CALL SOUND(100,660,0)
2190 PRINT :"* LOAD MODE"
2200 INPUT #3:A$,NAME$,PROD1
$,PROD2$,PROD3$
2210 CLOSE #3
2220 GOTO 2100
2230 CALL CLEAR
2240 OPEN #3:"CS1",INTERNAL,
OUTPUT,FIXED 192
2250 CALL SOUND(100,660,0)
2260 PRINT :"* SAVE MODE"
2270 PRINT #3:A$,NAME$,PROD1
$,PROD2$,PROD3$
2280 CLOSE #3
2290 GOTO 2100
2300 !
2310 STA$=SEG$(A$,1,2)
2320 IF STA$="00" OR STA$="0
1" OR STA$="02" OR STA$="03"
 OR STA$="04" OR STA$="05" O
R STA$="06" OR STA$="07" OR
STA$="08" OR STA$="09" THEN
ST$="USA/CANADA UPC" :: RETU
RN
2330 IF STA$="10" OR STA$="1
1" OR STA$="12" OR STA$="13"
 OR STA$="14" OR STA$="15" O
R STA$="16" OR STA$="17" OR
STA$="18" OR STA$="19" THEN
```

```
ST$="RESERVE......." :: RETU
RN
2340 IF STA$="20" OR STA$="2
1" OR STA$="22" OR STA$="23"
 OR STA$="24" OR STA$="25" O
R STA$="26" OR STA$="27" OR
STA$="28" OR STA$="29" THEN
ST$="INTERN NUMBERS" :: RETU
RN
2350 IF STA$="30" OR STA$="3
1" OR STA$="32" OR STA$="33"
 OR STA$="34" OR STA$="35" O
R STA$="36" OR STA$="37" THE
N ST$="FRANCE.....GEN" :: RE
TURN
2360 IF STA$="40" OR STA$="4
1" OR STA$="42" OR STA$="43"
 THEN ST$="W GERMANY  CCG" :
: RETURN
2370 IF STA$="49" THEN ST$="
JAPAN....." :: RETURN
2380 IF STA$="50" THEN ST$="
G BRITAIN  ANA" :: RETURN
2390 IF STA$="54" THEN ST$="
BELGIUM ICOF/D" :: RETURN
2400 IF STA$="57" THEN ST$="
DANMARK....EAN" :: RETURN
2410 IF STA$="64" THEN ST$="
FINLAND...CCCF" :: RETURN
2420 IF STA$="70" THEN ST$="
NORWAY....NKLF" :: RETURN
2430 IF STA$="73" THEN ST$="
SWEDEN...SEANC" :: RETURN
2440 IF STA$="76" THEN ST$="
SWITZERLD SACV" :: RETURN
2450 IF STA$="80" OR STA$="8
1" THEN ST$="ITALY  CCGICTI"
 :: RETURN
2460 IF STA$="84" THEN ST$="
SPAIN....." :: RETURN
2470 IF STA$="87" THEN ST$="
NETHERLAND UAC" :: RETURN
2480 IF STA$="90" OR STA$="9
1" THEN ST$="AUSTRIA....BAN"
 :: RETURN
2490 IF STA$="93" THEN ST$="
AUSTRALIA....." :: RETURN
2500 IF STA$="94" THEN ST$="
NEW ZEALAND..." :: RETURN
2510 IF STA$="98" OR STA$="9
9" THEN ST$="CREDIT NOTES  "
 :: RETURN
2520 !
2530 STA$=SEG$(A$,1,3)
2540 IF STA$="600" THEN ST$=
"SOUTH AFRICA  " :: RETURN
2550 IF STA$="859" THEN ST$=
"CSSR....." :: RETURN
2560 IF STA$="860" THEN ST$=
"YUGOSLAVIA...." :: RETURN
2570 IF STA$="729" THEN ST$=
"ISRAEL....." :: RETURN
2580 IF STA$="599" THEN ST$=
"HUNGARY....." :: RETURN
2590 ST$="--------------" ::
 RETURN
2595 FOR I=1 TO 12 :: PP$=SE
G$(A$,I,1):: IF PP$="P" THEN
 3010
2596 NEXT I
2600 PP$=SEG$(A$,13,1):: IF
PP$="P" THEN 2720
2610 ! PRUEFUNG
2620 PSUM=0
2630 FOR PX=1 TO 13
2640 PPP$=SEG$(A$,PX,1)
2650 PPP=VAL(PPP$)
2660 IF PX=2 OR PX=4 OR PX=6
 OR PX=8 OR PX=10 OR PX=12 T
HEN PSUM=PSUM+(3*PPP)ELSE PS
UM=PSUM+PPP
2670 NEXT PX
2680 PMOD=PSUM/10
2690 IF PMOD<>INT(PMOD)THEN
TTT$="NOT OK" ELSE TTT$=">>>
 OK"
2700 DISPLAY AT(5,21)SIZE(1)
:SEG$(A$,13,1)
2710 RETURN
2720 ! PRUEFZIFFER ERSTELLEN
2730 PSUM=0
2740 FOR PX=1 TO 12
2750 PPP=VAL(SEG$(A$,PX,1))
2760 IF PX=2 OR PX=4 OR PX=6
 OR PX=8 OR PX=10 OR PX=12 T
HEN PSUM=PSUM+(3*PPP)ELSE PS
UM=PSUM+PPP
2770 NEXT PX
2780 PMOD=PSUM/10 :: PQ=INT(
PMOD):: PREST=PMOD-PQ :: PZZ
=1-PREST
2790 IF PZZ=1 THEN PZZ$="0"
:: GOTO 2810
2800 PZZ1$=STR$(PZZ):: PZZ$=
SEG$(PZZ1$,2,1)
2810 DISPLAY AT(5,21)SIZE(1)
:PZZ$
2820 A$=SEG$(A$,1,12)&PZZ$
2825 TTT$="......"
2830 RETURN
2840 GOSUB 2310
2850 DISPLAY AT(5,1):"EAN":
:"STA": : :T1$: :T2$: :T3$:
:T4$: : :".....by DateContro
l": :!T5$
2860 DISPLAY AT(5,6):SEG$(A$
,1,2);" ";SEG$(A$,3,5);" ";S
EG$(A$,8,5);" ";SEG$(A$,13,1
)
2870 DISPLAY AT(7,6):ST$ ::
DISPLAY AT(10,6):NAME$ :: DI
SPLAY AT(12,6):PROD1$ :: DIS
PLAY AT(14,6):PROD2$ :: DISP
LAY AT(16,6):PROD3$
2880 DISPLAY AT(2,1):"~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~" :: D
```

LISTING

```
ISPLAY AT(23,1):"!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!"
2890 GOTO 2990
2900 ACCEPT AT(5,6)SIZE(-13)
VALIDATE("0123456789P")BEEP:
A$
2910 DISPLAY AT(5,6):SEG$(A$
,1,2);" ";SEG$(A$,3,5);" ";S
EG$(A$,8,5);" ";SEG$(A$,13,1
)
2920 GOSUB 2595 :: DISPLAY A
T(5,6)SIZE(2):SEG$(A$,1,2)::
 DISPLAY AT(5,9)SIZE(5):SEG$
(A$,3,5):: DISPLAY AT(5,15)S
IZE(5):SEG$(A$,8,5)
2930 GOSUB 2310 :: DISPLAY A
T(5,23):TTT$ :: DISPLAY AT(7
,6):ST$ :: DISPLAY AT(7,23):
"GOON Y" :: ACCEPT AT(7,28)S
IZE(-1)VALIDATE("YN")BEEP:Y$
2940 IF Y$="N" THEN 3010 ELS
E IF Y$="Y" THEN 2950 ELSE 3
010
2950 ACCEPT AT(10,6)SIZE(-23
)BEEP:NAME$
2960 ACCEPT AT(12,6)SIZE(-23
)BEEP:PROD1$
2970 ACCEPT AT(14,6)SIZE(-23
)BEEP:PROD2$
2980 ACCEPT AT(16,6)SIZE(-23
)BEEP:PROD3$
2990 DISPLAY AT(24,1):"PRESS
  N/EW P/RINT M/ENUE  N" ::
ACCEPT AT(24,28)SIZE(-1)VALI
DATE("NPM")BEEP:PP$
3000 IF PP$="P" THEN 230 ELS
E IF PP$="M" THEN 2040 ELSE
IF PP$="N" THEN 3010 ELSE 32
767
3010 DISPLAY AT(5,6)SIZE(16)
:"" :: DISPLAY AT(5,6)SIZE(1
3):A$
3020 GOTO 2900
```

Fortsetzung von Seite 115

puter ermitteln zu lassen. Geben Sie dafür bei "EAN" 12 numerische Stellen sowie ein "P" für Prüfziffer ein. Z.B. 40123456789P, der Computer ermittelt bei diesem Beispiel die 1 als Prüfziffer und integriert diese. Der weitere Ablauf des Programmes ist mit dem unter <A> identisch.
Drücken Sie jetzt "P".
Das Ergebnis spricht für sich.
Am Schluß sei noch einmal auf die ausschließlich private Nutzung dieses Programmes hingewiesen.

Thomas Jerusalem

BÖRSE

Verk. TI 99/4A weg. Systemwechsel. Konsole u. Ex-Basic nur 350,–. Jedes Spiel nur 1,–. Listings nur –,20 DM pro Seite. 32 KB u. Disklaufw. je 225,– DM. Info gratis von: Markus Theimer, Schwedenstr. 2, 6203 Hochheim a. M.

Verk. Zusatzinfo für E/A Handbuch. Mit o. ohne Disk. sehr ausführl. (Bitdarst.) u. einfache Bsp. Ideal für Einstieg in Assembler. Keine Vorkenntn. nötig. Handbuch genügt. Ab Mitte Juli lieferb. Probeseiten b. Rückporto mögl. Andreas Pack, Eickelerstr. 60, 4690 Herne 2

Verk. defekte TI-Konsole 120,–, Joystickadapter 15,–, Palmodulator 40,–, Adapter 35,–, Spiele 25,–, Orig. Kass.Rek. von TI 70,–, Ext. Basic-Modul 100,–. Melden bei Thorsten Löw, Tel: 07130/7165

Achtung! TI-Freaks! Suche dringend Ex-Basic, aber nur dt. Nachbau! Und Expansion Box m. Karten (DSK-Laufw. usw.) u. Speechsynthesizer u. ext. 32 K-Erw. m. Centr. interface! Angebote an: Tel: 0201/712816

Programmiere Ihre Eproms: z.B. für Modulplatine, Q-Save-Modul. Suche günstig: Sprachsynthesizer, Modulgehäuse, Angebote an und Info von: Peter Rieger, Talstr. 64, 6750 Kaiserslautern. Tel: 0631/73669

Suche Ex-Basic (dt. Handbuch) zu günst. Preis. Tel: 02333/4343, ab 16 Uhr

---

Atronic V. 24 Schnittstelle für 200,– DM abzugeben. Grund: An durchgeführt. BUS der Schnittstelle angeschlossene Peripherie funk. nicht. R. Arens, Saazer Weg 1, 3170 Gifhorn

---

Verk. fast neues TI 99/4A System – nur kompl. – (inkl. 2 CPU, 2 Rec., TI-2-Rec.-Kabel, 1 Exp. Unit mit RS 232-Karte, 32 KB Erw., Disk-Contr., int. Laufwerk, TI-Matrixprinter, Speech-Synt., TI-Joysticks), div. TI-orig. Software (TI-Ass., TI-Multiplan, TI-Writer, TI-Rechnungsst., TI-Buchh., TI-Ext. Basic, TI-Schach u.v.a.), TI-Lit. Dr. Hanne, Im Langen Feld 16, 3014 Laatzen 1, Tel: 0511/821614

Verk. od. tausche Progr. Listen anfordern bei: Andreas Tilch, Weierstadt 17, 6648 Wadern-Büschfeld/Saar, Tel: 06874/6705

Progr. zur Erstellung von Lotto-Syst. f. 6 aus 49 mit Gewinngarantie. Ex-Basicprogr. auf Kass. geg. Einsend. von 10,– DM. P. Hielscher, Am Wall 22, 4401 Saerbeck

---

TI-User: Laßt Euch von Geschäftemachern nicht ausnehmen! Bei uns gibt's US-Beginners Basic Tutor für nur 6,– DM inkl. Porto. Tel: 06638/1503, ab 15 Uhr

---

Wer hat Software für das Mini-Mem. Erwin Kinslechner, Eibesbrunnergasse 1/10 A-1120 Wien

---

TI99/4A Konsole m. eingeb. 32K-CMOS Erw. zu verk. für 250 DM. Rechner ist z.B. in Ass. bis zu 2 x so schnell, dß diese Erw. wie die CPU auch einen echten 16Bit Datenbus besitzt. Tel. 06257/83247

Suche gegen Bezahlung: Bedienungsanleitung f. TI (auch Fotokopie) + Progr. Bildschirmausdruck (Hardcopy) f. Drucker Kaitec 100 Ex. Suche außerdem: Peribox m. 2 Disclw. + Control + RS232 Schnitt stellenkarte. Tel. 07026/5885

LISTINGS

```
10 ! **********************
11 ! *                    *
12 ! *  WAEHLERVERHALTEN  *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *    Hans Enders     *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     1666 Bytes     *
28 ! *                    *
29 ! **********************
100 !CALL SCREEN(10)
110 CALL CLEAR
120 F$="FFFFFFFFFFFFFFFF" ::
 D$="FF7F3F1F0F0F0301" :: V$
="FF818181818181FF"
130 CALL CHAR(96,V$,104,V$,1
12,V$):: CALL COLOR(9,7,16,1
0,2,11,11,16,6)
140 FOR Z=3 TO 20
150 FOR S=5 TO 23
160 RANDOMIZE
170 K=INT(3*RND)+1
180 IF K=1 THEN F=96 ELSE IF
 K=2 THEN F=112 ELSE F=32
190 CALL HCHAR(Z,S,F,1)
200 NEXT S
210 NEXT Z
220 P=INT(8*RND)+1
230 Z=INT((20-3+1)*RND)+3
240 S=INT((23-5+1)*RND)+5
245 CALL GCHAR(Z,S,K1):: IF
K1=32 THEN 220
250 A=Z+1
260 B=Z-1
270 C=S+1
280 D=S-1
290 IF A>=20 THEN A=3
300 IF B<=3 THEN B=20
310 IF C>=23 THEN C=5
320 IF D<=5 THEN D=23
340 IF P=1 THEN CALL GCHAR(B
,S,K)
350 IF P=2 THEN CALL GCHAR(B
,C,K)
360 IF P=3 THEN CALL GCHAR(Z
,C,K)
370 IF P=4 THEN CALL GCHAR(A
,C,K)
380 IF P=5 THEN CALL GCHAR(A
,S,K)
390 IF P=6 THEN CALL GCHAR(A
,D,K)
400 IF P=7 THEN CALL GCHAR(Z
,D,K)
410 IF P=8 THEN CALL GCHAR(B
,D,K)
411 ON P GOSUB 1000,1020,104
0,1060,1080,1100,1120,1140
440 !CALL HCHAR(Z,S,K,1)
450 DISPLAY AT(24,5):"ZEIT="
;N
460 N=N+1
470 GOTO 220
1000 IF K<>K1 THEN CALL HCHA
R(B,S,K1,1):: IF K=32 THEN C
ALL HCHAR(Z,S,32,1)ELSE CALL
 HCHAR(Z,S,K1,1)
1010 RETURN
1020 IF K<>K1 THEN CALL HCHA
R(B,C,K1,1):: IF K=32 THEN C
ALL HCHAR(Z,S,32,1)ELSE CALL
 HCHAR(Z,S,K1,1)
1030 RETURN
1040 IF K<>K1 THEN CALL HCHA
R(Z,C,K1,1):: IF K=32 THEN C
ALL HCHAR(Z,S,32,1)ELSE CALL
 HCHAR(Z,S,K1,1)
1050 RETURN
1060 IF K<>K1 THEN CALL HCHA
R(A,C,K1,1):: IF K=32 THEN C
ALL HCHAR(Z,S,32,1)ELSE CALL
 HCHAR(Z,S,K1,1)
1070 RETURN
1080 IF K<>K1 THEN CALL HCHA
R(A,S,K1,1):: IF K=32 THEN C
ALL HCHAR(Z,S,32,1)ELSE CALL
 HCHAR(Z,S,K1,1)
1090 RETURN
1100 IF K<>K1 THEN CALL HCHA
R(A,D,K1,1):: IF K=32 THEN C
ALL HCHAR(Z,S,32,1)ELSE CALL
 HCHAR(Z,S,K1,1)
1110 RETURN
1120 IF K<>K1 THEN CALL HCHA
R(Z,D,K1,1):: IF K=32 THEN C
ALL HCHAR(Z,S,32,1)ELSE CALL
 HCHAR(Z,S,K1,1)
1130 RETURN
1140 IF K<>K1 THEN CALL HCHA
R(B,D,K1,1):: IF K=32 THEN C
ALL HCHAR(Z,S,32,1)ELSE CALL
 HCHAR(Z,S,K1,1)
1150 RETURN
```

WÄHLER-VERHALTEN

Wähler entstammt aus einer Anregung aus der Computer-Kurzweil im Spektrum der Wissenschaft. Auf einer torusförmigen Welt, richtet sich das Wahlverhalten eines Einwohners nach dem Verhalten eines seiner Nachbar. Wenn es sich um ein leeres Feld handelt, so wandert der Einwohner dorthin. Andernfalls, wenn der Nachbar nicht seiner Partei angehört, so nimmt der Nachbar die Farbe seines Nachbarn an.
Das Programm kann man stundenlang beobachten und Änderungen feststellen. Sicher ist nur eines: Nach vielen Stunden ist die Demokratie einer Diktatur gewichen.

Bogen 2

VERDIENEN SIE GELD MIT IHREM COMPUTER!

Haben Sie einen TI 99/A? Können Sie programmieren? In Basic oder Maschinensprache? Dann bietet TI – REVUE Ihnen die Möglichkeit, damit Geld zu verdienen.

Wie? Ganz einfach. Sie senden uns die Programme, die Sie für einen Abdruck als geeignet halten, zusammen mit einer Kurzbeschreibung, aus der auch die verwendete Hardware – eventuelle Erweiterungen – benutzte Peripherie – hervorgehen muß (Schauen Sie sich dazu den Kopf unserer Programmlistings an.)

Benötigt werden: Zwei Listings des Programms sowie eine Datenkassette oder Diskette! Wenn die Redaktion sich überzeugt hat, daß dieses Programm läuft und sich zum Abdruck eignet, zahlen wir Ihnen pro Programm je nach Umfang bis zu DM 300,–!

Sollten Sie keinen Drucker haben, genügt der Datenträger.

Sie erhalten Ihre Kassette/Diskette selbstverständlich zurück, wenn Sie einen ausreichend frankierten Rückumschlag mit Ihrer Adresse beifügen.

Bei der Einsendung müssen Sie mit Ihrer Unterschrift garantieren, daß Sie der alleinige Inhaber der Urheber-Rechte sind! Benutzen Sie bitte anhängendes Formular! (Wir weisen darauf hin, daß auch die Redaktion amerikanische und englische Fachzeitschriften liest und „umgestaltete" Programme ziemlich schnell erkennt).

Um Ihnen die Arbeit zu erleichtern, finden Sie hier ein Formular. Sie können es ausschneiden oder fotokopieren.

PROGRAMMANGEBOT

Name des Einsenders: _____

Straße/Hausnr./Tel.: _____

Plz/Ort: _____

Hiermit biete ich Ihnen zum Abdruck folgende(s) Programm(e) an:

_____

_____

_____

_____

Benötigte Geräte: _____

_____

Beigefügt ( ) Listings ( ) Kassette ( ) Diskette

Ich versichere, der alleinige Urheber des Programmes zu sein!
Hiermit ermächtige ich die Redaktion, dieses Programm abzudrucken und wirtschaftlich zu verwerten. Sollte es in den Kassetten-Service aufgenommen werden, erhalte ich auch dafür eine entsprechende Vergütung, das Copyright geht auf den Verlag über.

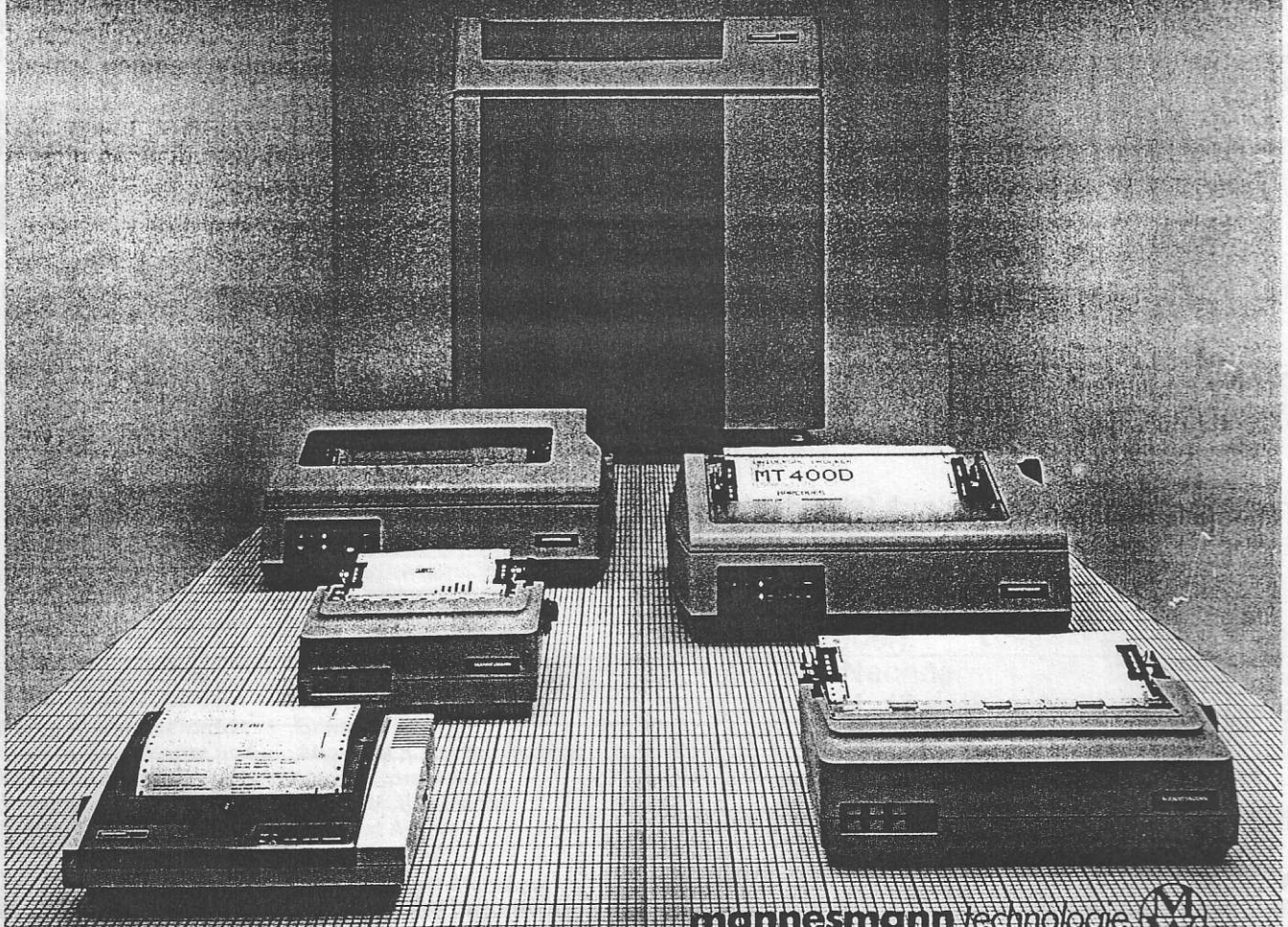
_____

Rechtsverbindliche Unterschrift

TI-REVUE
Postfach 1107
8044 Lohhof

MANNESMANN
TALLY

Computerperipherie der Mannesmann-Tally-Klasse



mannesmann *technologie*

Drucker in jeder Leistungsklasse, zwischen 100 Zeichen/Sekunde und mehr als 600 Zeilen/Minute Druckgeschwindigkeit, für Home Computer, Personal Computer, Bürocomputer, Textsysteme, EDV-Systeme. Drucker mit vielen Zusatzausstattungen wie Einzelblattzufuhr, Stapelzufuhr, Mehrfarbdruck, Etikettendruck oder Schneidevorrichtung. Drucker mit vielen, wählbaren Schriftarten.

Schnellschrift, Schönschrift, OCR-Schrift, Plakatschrift oder Barcode.

Mannesmann-Tally-Klasse – das sind erfolgreiche Drucker, kompatibel zu erfolgreichen Computersystemen und dazu der qualifizierte, kundennahe Service.

Mannesmann Tally GmbH
Böblinger Str. 10, 7000 Stuttgart 50
Telefon 07 11 / 5 03 90, Telex 7 254 672

1133

This collection is for all TI99/4A user.
If you like TI99 join International Forum present on AtariAge site.
Greetings from Italy. **Astharot**